



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

ENGENHARIA DA COMPUTAÇÃO

**SARCO – SISTEMA AUTOMÁTICO DE
RECONHECIMENTO E CONTAGEM DE
OVOS DA DENGUE**

Trabalho de Graduação

Aluno: Styve Stallone da Silva(sss4@cin.ufpe.br)

Orientador: Aluizio Araújo Fausto Ribeiro
(*aluizioa@cin.ufpe.br*)

Recife, julho de 2010

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
ENGENHARIA DA COMPUTAÇÃO

**SARCO – SISTEMA AUTOMÁTICO DE
RECONHECIMENTO E CONTAGEM DE
OVOS DA DENGUE**

Trabalho de Graduação

Monografia apresentada à
Universidade Federal de Pernambuco,
como trabalho de conclusão do curso
de Graduação em Engenharia da
Computação, para a obtenção do
título de Engenheiro da Computação.

Aluno: Styve Stallone da Silva(sss4@cin.ufpe.br)

Orientador: Aluizio Araújo Fausto Ribeiro (*aluizioa@cin.ufpe.br*)

Recife, Julho de 2010

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

ENGENHARIA DA COMPUTAÇÃO

**SARCO – SISTEMA AUTOMÁTICO DE
RECONHECIMENTO E CONTAGEM DE
OVOS DA DENGUE**

Trabalho de Graduação

Monografia apresentada à Universidade Federal de Pernambuco, como trabalho de conclusão do curso de Graduação em Engenharia da Computação, para a obtenção do título de Engenheiro da Computação.

Aprovada em ____ de _____ de _____.

Prof. _____

Prof. _____

(Orientador)

Recife, Julho de 2010

"Um homem é um sucesso se pula da cama de manhã e vai dormir à noite, e nesse meio tempo faz o que gosta."

Bob Dylan

AGRADECIMENTOS

A Deus em primeiro lugar. Minha Família. Painho, que me incentivou e me **sustentou** todos esses anos de graduação, e muito mais, ele ensinou muitas coisas importantes em toda a vida que ajudaram a formar o meu caráter. Mainha sempre paciente com meus estresses em vésperas de provas e deadlines de projetos. Minha tia e madrinha, Fátima, segunda mãe – acho que não preciso mais falar nada. Meu primo Anderson, que maluco feito eu, vai se formar Engenheiro da Computação junto comigo, espero! e que me ajudou aos longos desses anos da graduação. A todos os meus primos que conviveram comigo nesses anos e puderam me ajudar e curtir o que pouco me sobrou de tempo: gugu, André, Rominho, Robinho, Cleyton, Sting (Lôlô), Eline, Eliezer. Todos os tios e tias – fuliões ou não, e meus avôs.

Aos companheiros de curso. Os futuros Engenheiros: Adelmo (pois eh!), Silvio Santos vem aí, LALA..LALA, Igino, Lúcio, TAVL(calmo!), Biu(o grande!), J.P., Libório, Bruno BPN, Theogenes, Cirdes, Ciro(29”), Hugo, REAF, Apebão, Ângelo (Jesus!) entre outros. Apesar de serem fuliões, grandes companheiros nas provas, projetos e farras. Aos cientistas: Allan, Armando, Flávio Almeida, João Paulo(Macaxeira), João Gabriel(Vanessão), Lucas Araújo, Arara, Leo Vilaça e um monte mais.

Aos companheiros de projeto e laboratório: André de Melo, Daniel, Orivaldo(que apesar de não ser do projeto me ajudou muito), Flávia, Renata, Cícero, Zé Henrique e principalmente André Tiba e Hansenclever Bassani que me ajudaram muito durante o desenvolvimento do projeto com suas experiências, críticas e paciência em ensinar o que eu não sabia.

Aluízio, meu orientador de IC e TG, que apesar de muita ocupação estava sempre tentando ajudar a obtermos os melhores resultados, e me ajudou a concluir o curso.

A todos os professores que puderam ajudar na minha formação. Todos os funcionários do Cin que de alguma forma nos proporcionaram um ambiente digno.

Aos amigos presentes em alguns desses anos: Léo, Gilson, André Gordo, Karla, Rebeca, Lorena, Gracilene, Aline, Gabriela, Mayara, Lon e todos meus companheiros do motoclub.

RESUMO

Atualmente há um grande empenho do governo em criar projetos e realizar ações que evitem a proliferação do mosquito *Aedes aegypti*, com objetivo de evitar a transmissão da dengue, doença que vem afetando grande parte da população pernambuca e brasileira.

A secretaria de saúde vem realizando um trabalho de contagem dos ovos da dengue, para fazer o mapeamento das áreas onde há maior infestação. Porém esse trabalho envolve um grande custo financeiro e muito desgaste físico. A contagem é realizada, com o auxílio de microscópio, através de armadilhas em madeira espalhadas pela cidade onde os ovos são depositados. O projeto desenvolvido visa realizar a contagem desses ovos de forma automática.

Será desenvolvido um sistema neural no qual essas armadilhas, chamadas de ovitrampas, serão digitalizadas com *scanner* e as informações das suas imagens serão passadas ao sistema, que realizará de forma automática a contagem dos ovos em cada ovitrampa.

Palavras-chave: Redes Neurais, RBF

Lista de Figuras

Figura 1. Esquema de unidade McCullock – Pitts.	16
Figura 2. MultiLayer de uma camada escondida.	17
Figura 3. MultiLayer com duas camadas escondidas.....	17
Figura 4. Segurança, Realidade Aumentada e Robótica, respectivamente.	18
Figura 5. Uma possível superfície de erro com os mínimos locais e o mínimo global demonstrada graficamente	21
Figura 6. Rede RBF com uma saída	23
Figura 7. Partição de dados em clusters	25
Figura 8. Esquema do treinamento supervisionado.	28
Figura 9. Passo direto do backpropagation.....	30
Figura 10. Passo reverso do backpropagation.	30
Figura 11. Esquema do treinamento não supervisionado.	31
Figura 12. Ilustração do k-médias.	32
Figura 13. Gráfico da Função Sigmóide.....	35
Figura 14. Estrutura do SARCO, com seus diversos módulos de processamento.....	36
Figura 15 - Imagem de um ovos e um conjunto de máscaras quadrada e circular (em vermelho) utilizadas para descrever um pixel da imagem (que está sob o ponto amarelo no centro da máscara vermelha quadrada).	41
Figura 16 - Imagens da 1ª linha (com ovos) e 3ª linha (sem ovos, apenas ruído) foram utilizadas para criar o conjunto de treinamento. Imagens centrais são as segmentações manuais das imagens da 1ª linha, ou seja, são as rotulações de todos os pixels das imagens da 1ª linha.	42
Figura 17 - esquerda superior: imagem de um bloco; direita superior: segmentação manual; esquerda central: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS em amarelo); direita central: segmentação com RBF treinada com o Conjunto C2; esquerda inferior: segmentação com RBF treinada com o Conjunto C2 _{repr}	46
Figura 18 – esquerda superior: bloco sem ovos e com manchas; direita superior: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS pintados em amarelo); esquerda inferior – segmentação com RBF treinada com o Conjunto C2; direita inferior – segmentação com RBF treinada com o Conjunto C2 _{repr}	47
Figura 19 - esquerda superior: bloco da borda da palheta, excluída na Seleção ₁ de blocos; direita superior: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS em amarelo); esquerda inferior: segmentação com RBF treinada com o Conjunto C2; direita inferior: segmentação com RBF treinada com o Conjunto C2 _{repr}	48
Figura 20 - esquerda superior: bloco com mancha, excluído na Seleção ₂ de blocos; direita superior: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS em amarelo); esquerda inferior: segmentação com RBF treinada com o Conjunto C2; direita inferior: segmentação com RBF treinada com o Conjunto C2 _{repr}	49
Figura 21 - Ajustes linear e quadrático para os dados de segmentação manual de 9 palhetas (998 blocos).	50

Lista de Tabelas

Tabela 1 - Resultados das simulações com Rede Neural RBF para os dados de treinamento e validação, para os três conjuntos de dados CD1, CD2 e CD2 _{repr}	45
Tabela 2 - erros quadráticos médios dos melhores ajustes sobre os dados de segmentação manual, para três tipos de seleção de blocos.	51
Tabela 3 - Estimativas do número de ovos nas palhetas simuladas com uma rede RBF para palhetas descritas pelo conjunto de C1, com a seleção de blocos Seleção ₂	52
Tabela 4 - Estimativas do número de ovos nas palhetas simuladas com uma rede RBF, com o conjunto de dados C2, com a seleção de blocos Seleção ₂	53
Tabela 5 - Estimativas do número de ovos nas palhetas simuladas com uma rede RBF para palhetas descritas pelo conjunto de dados C2 _{repr}	54
Tabela 6 - mesmo que a Tabela 53, porém todos os blocos foram selecionados.	55

Sumário

Lista de Figuras	7
Lista de Tabelas	8
1. Introdução	10
1.1 Motivação.....	10
1.2 Objetivo	10
1.3 Visão Geral do Trabalho	10
2. O problema.....	11
3. Redes Neurais.....	13
3.1 Histórico	13
3.2 Motivação para Uso de RNA's.....	14
3.2 Características Gerais	16
3.4 Aprendizagem	18
3.4.1 Aprendizagem por Correção de Erro.....	20
4. RBF.....	23
4.1 Características da RBF comparadas com a MLP.....	25
4.2 Funcionamento da RBF no SARCO	26
4.3 Aprendizagem Supervisionada.....	27
4.3.1 <i>BackPropagation</i>	28
4.4 Aprendizagem Não-supervisionada	30
4.4.1 Algoritmo do <i>k-médias</i> e seu uso no Sarco	31
4.5 Função de Ativação da Camada de Saída	34
5. Proposta do SARCO	36
5.1 Função de Base Radial.....	38
5.2 Aquisição das Imagens	38
5.3 Extração de Características das Imagens das Palhetas	40
6. Problemas na Implementação, Solução para os Problemas, Testes e Resultados	43
6.1 Resultados	43
6.2.1 Estimativa do Número de Ovos.....	50
6.2.2. RBF.....	51
6.3 Problemas e soluções.....	56
7. Conclusões e Trabalhos Futuros.....	57
8. Referências Bibliográficas	59

1. Introdução

1.1 Motivação

O desgastante trabalho de contagem dos ovos da dengue, realizado pela secretaria, tem como objetivo avaliar o grau de infestação de uma determinada área. Ela espalha por diversas áreas da cidade pequenas placas, chamadas de ovitrampas, onde o mosquito deposita seus ovos. A partir dessas placas, técnicos de saúde realizam a contagem dos ovos com o auxílio do microscópio. Um dos problemas é que o custo desses técnicos é alto e outro é que essa contagem realizada através do microscópio causa um desgaste na vista dos técnicos devido ao alto tempo que eles passam contando durante o dia. Visando resolver ou amenizar esses problemas está sendo desenvolvido um sistema de contagem automatizado desses ovos.

1.2 Objetivo

Para solucionar os problemas da contagem manual foi proposto um sistema de reconhecimento e contagem automático. O sistema desenvolvido será composto por uma rede neural RBF (*Radial Basis Function*). O sistema terá como entrada os dados das imagens das ovitrampas digitalizadas e partir delas realizará o reconhecimento e contagem dos ovos de forma automatizada.

1.3 Visão Geral do Trabalho

Este trabalho será dividido em sete capítulos. O capítulo 2 descreverá o problema a ser solucionado e as soluções já existentes. O capítulo 3 falará sobre as redes neurais artificiais. O capítulo 4 mostrará as características das redes RBF. O capítulo 5 apresentará a proposta do SARCO (Sistema Automático de Reconhecimento e Contagem de Ovos da Dengue). O capítulo 6 mostrará os problemas do projeto, soluções encontradas, os testes realizados e os resultados obtidos. O capítulo 7 trará as conclusões e trabalhos futuros.

2. O problema

Desde o surgimento e a disseminação da dengue, o governo vem fazendo grandes esforços para controlar a proliferação do seu vetor transmissor, o mosquito *Aedes aegypti*. Tem-se espalhado gases pelas ruas e tem-se feito um esforço enorme de tentar conscientizar a população a não dar ao mosquito o ambiente que ele necessita para a reprodução. Porém, é bastante difícil fazer toda a população tomar consciência dos problemas da doença e tentar eliminar o mosquito. Com todos esses problemas a rede Saudável, da qual faz parte o Aggeu Magalhães (CPqAM), que representa a FioCruz no estado de Pernambuco, criou um Sistema de Monitoramento e Controle Populacional do Vetor da Dengue (SMCP-Aedes)[27].

A Secretaria de Saúde do estado de Pernambuco utilizou em alguns bairros do Recife esse sistema, que tem como objetivo retirar ovos do mosquito *Aedes aegypti* do meio ambiente, para fazer a contagem e conseqüentemente, mapear as áreas mais infestadas pelo mosquito. Para a realização desse monitoramento, o sistema conta com uma armadilha chamada de ovitrampa. Ela é composta de um pequeno balde com três palhetas de eucatex, larvicida biológico, e alfafa, que serve pra atrair a fêmea do mosquito. Após algum tempo essas palhetas são substituídas e as antigas levadas ao laboratório para a contagem.

O sistema de contagem dos ovos nas palhetas recolhidas das armadilhas era feito antes pela secretaria de modo totalmente manual. Quando o sistema foi expandido para o interior do estado, a secretaria apoiou o desenvolvimento de um sistema semi-automático de contagem, o SCAC (Sistema de Contagem Assistida pelo Computador) desenvolvido pelo Centro de Informática da Universidade Federal de Pernambuco. Nesse sistema, a palheta é digitalizada e um técnico faz a contagem utilizando o SCAC. Porém esse sistema apesar de já ter proporcionado uma melhora no sistema de contagem, pois já não era mais necessário o uso do microscópio, o que tornava o trabalho bem menos desgastante, ainda dependia do técnico para realizar a contagem em cada imagem.

Com isso foi proposto um sistema automático de contagem dos ovos. Essa proposta levou ao desenvolvimento do SARCO.

A idéia principal desse monitoramento para posterior mapeamento é como citado acima facilitar a tentativa de diminuir a reprodução do mosquito. Os trabalhos realizados com a população podem ser melhor direcionados quando a secretaria sabe quais as áreas que mais apresentam focos de reprodução do mosquito. Por exemplo, a ação do uso de gases pode ser melhor aproveitado, ou seja, só receberá esse serviço as áreas onde sejam realmente necessárias, sem que haja desperdício de serviço, tempo e dinheiro.

3. Redes Neurais

Redes Neurais artificiais são sistemas baseados no sistema cerebral humano. São sistemas que tem como características aprender, adaptar-se e tirar conclusões inspirando-se no funcionamento do cérebro humano[5]. Tais redes se fundamentam em ligações entre unidades, chamadas de nós simples, e formam assim uma rede de nós. Esses nós enviam ou recebem, através das ligações, sinais para os outros nós; os nós simulam os neurônios cerebrais e as ligações entre eles as sinapses.

3.1 Histórico

Os primeiros estudos foram divulgados em 1943 por McCulloch e Pitts, eles falaram de uma máquina que teria inspiração no cérebro humano[3]. Em 1951 teve-se também o desenvolvimento do primeiro computador neural, o *Snark*, desenvolvido por Marvin Minsky. O *Snark* ainda simples operava com pesos ajustados automaticamente, e serviu de inspiração para estruturas que foram produzidas posteriormente.

Em 1957 e 1958, Frank Rosenblatt, Charles Wightman e outros desenvolveram o Mark I *Perceptron*, o primeiro modelo de rede neural a ter repercussão. Após o feito de Rosenblatt, Bernard Widrow desenvolveu o Adaline, novo tipo de processamento de redes neurais, que tem uma regra de aprendizagem poderosa e ainda hoje se mantém em uso.

Devido a problemas de muito entusiasmo e pouca prova sobre máquinas tão poderosas quanto o cérebro humano que surgiram nesse tempo, o estudo dessa área se tornou pouco confiável e poucas pesquisas foram feitas nos anos seguintes. Aqueles que continuaram pesquisando, o fizeram em particular até conseguir encontrar argumentos concretos para que voltassem a ter um grande estudo nesse assunto.

Só a partir de meados dos anos 80, os pesquisadores voltaram a publicar propostas para o desenvolvimento de redes neurais e de como utilizá-las. Em 1983 o líder da *DARPA (Defense Advanced Research Projects Agency)*, Ira Skurnick, deu o passo mais importante desta época, ele

fundou pesquisas em neurocomputação ouvindo o que os pesquisadores estavam falando sobre isso. Com isso a *DARPA* se tornou uma líder mundial em inovação tecnológica e abriu o caminho para a neuro computação [3]. Nessa mesma época, um grande físico, John Hopfield, interessou-se pela área e escreveu artigos que conseguiram fazer cientistas e tecnólogos a se interessarem também pela área e começar a estudá-la.

Em 1986 David Rumelhart e James McClelland publicaram o livro *Parallel Distributed Processing*, fazendo com que o estudo dessa área ganhasse proporções grandiosas. Em 1987 em São Francisco foi realizada a primeira conferência de redes neurais em tempos modernos, a *IEEE International Conference on Neural Networks*. E a partir de 1987 universidades começaram a criar centros de pesquisa em neurocomputação [3].

3.2 Motivação para Uso de RNA's

Em um certo momento da história, com a evolução das máquinas, o homem começou a ter o desejo de criar máquinas que pudessem operar de maneira autônoma. Era desejada uma máquina que tivesse capacidade de aprender a interagir com um ambiente desconhecido a ela.

O sucesso dessas máquinas autônomas depende exclusivamente da capacidade que elas terão de se adaptar ao ambiente que a envolve. Com essa capacidade de adaptação elas poderiam continuar a realizar as tarefas mesmo sem conhecer as possíveis mudanças do ambiente, algo que não aconteceria com as máquinas atuais. O uso dessas máquinas é de grande importância onde a atividade humana é perigosa, impossível ou tediosa, como operações militares, exploração do espaço onde a distância impediria o controle de uma nave [3].

As máquinas atuais possuem um processamento baseado em modelos matemáticos, já as máquinas baseadas na inspiração biológica, utilizam cálculos matemáticos para realizar suas operações, porém são livres ao ponto que precisarem para executar tarefas em ambientes desestruturados.

Os sistemas de computação baseados em redes neurais conseguem receber ao mesmo tempo diversas entradas e utilizá-las de maneira organizada. Uma grande vantagem que torna as redes neurais sistemas redundantes a falhas, é que diferentemente de sistemas atuais que possuem armazenamento baseados em memória, ou seja, uma informação fica guardada em um determinado endereço, caso ele seja danificado a informação é perdida, as informações numa rede neural é guardada em todas as suas unidades de processamento, isto é, caso falhe uma parte da rede ou seja destruída, ainda tem-se a mesma informação em outro elemento.

A característica muito importante na estrutura das redes neurais é sua habilidade de aprendizagem e adaptação. Devido a essas habilidades, com um treinamento bem feito a rede consegue generalizar seu funcionamento para entradas desconhecidas por ela, e consegue também lidar com situações de indefinidas e imprecisão de dados.

A principal característica de uma rede neural é definida pela habilidade que ela possui em fazer a aproximação de qualquer função contínua não-linear. Com isso é possível modelar sistemas não-lineares na combinação de componentes não-lineares [3].

A propriedade do mapeamento de entrada-saída é utilizada na aprendizagem supervisionada. Nesse treinamento os pesos sinápticos são alterados com amostras de um conjunto de treinamento. Um exemplo é escolhido ao acaso e possui um sinal de entrada e uma resposta desejada, esse sinal é usado para treinar a rede e tentar minimizar a diferença entre a saída da rede e a resposta desejada. Esse treinamento é feito até não haver mais nenhuma mudança significativa dos pesos sinápticos da rede [21].

A rede neural obedece à propriedade chamada de Resposta a Evidência. Essa propriedade permite a rede além de classificar um padrão, dar a informação do quão confiável é a sua classificação, podendo dessa forma rejeitar padrões ambíguos, se existirem, e assim melhorar o desempenho da rede [21].

As redes possuem Uniformidade de Análise e Projeto, ou seja, as redes utilizam a mesma notação em todas as aplicações. Verifica-se essa característica de diversos modos [21]:

- Os neurônios, representam um ingrediente comum a todas as redes.
- A uniformidade permite que seja utilizada as mesmas teorias em aplicações diferentes de redes neurais.

3.2 Características Gerais

As redes podem ser desde modelos mais simples, como o *Perceptron*, que só possui duas camadas, a de entrada e a de saída, até modelos com várias camadas. O modelo do *Perceptron*, devido à inexistência de camada intermediária, consegue resolver apenas problemas linearmente separáveis. Modelos com múltiplas camadas, conhecido com MLP (*Multilayer perceptron*) podem resolver os demais problemas. Se o MLP tiver apenas uma camada intermediária essa rede consegue resolver qualquer função contínua, se tiver mais de uma ela resolve todas as funções, contínua ou não [5]. Esquemas do MLP de uma camada e de mais camadas estão representados da figura 2 e na figura 3 respectivamente.

O modelo proposto por McCulloch e Pitts, o *Perceptron*, se resume a sinais de entrada, no qual cada um desse sinal é multiplicado por seu peso e com isso temos uma soma ponderada das entradas. Este valor é comparado a um limiar e uma saída é produzida [15]. O esquema do perceptron é mostrado na figura 1 abaixo.

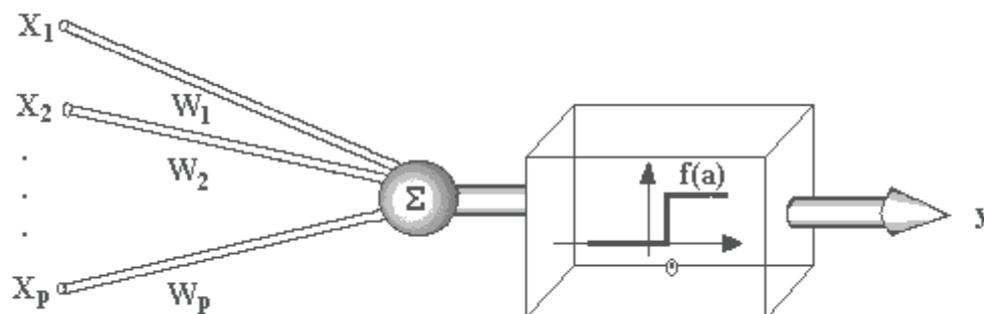


Figura 1. Esquema de unidade McCulloch – Pitts.

Esse esquema mostra o que está descrito acima, teremos a saída da seguinte, mostradas na Equação 1 e Equação 2 abaixo.

$$Y = 1, \text{ se } a \geq t$$

Equação 1. saída da rede

$$Y = 0, \text{ se } a \leq t$$

Equação 2. saída da rede

no qual $a = W_1X_1 + W_2X_2 + \dots + W_PX_P$, e $t = \text{limiar}$.

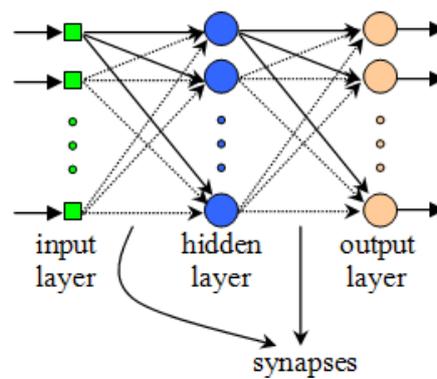


Figura 2. MultiLayer de uma camada escondida.

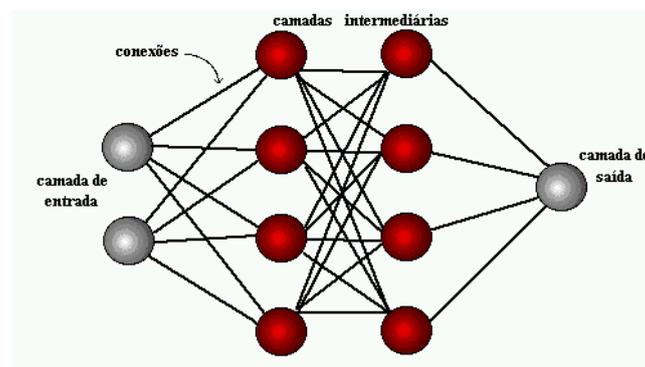


Figura 3. MultiLayer com duas camadas escondidas.

O uso das redes neurais vem sendo a cada dia mais disseminado nas mais diversas aplicações, sejam elas aplicações críticas para o ser humano

ou apenas para melhorar o desempenho de alguma atividade. Alguns exemplos são mostrados na figura 4. Na primeira imagem, por exemplo, temos um sistema de segurança, onde o sistema faz a detecção de movimento e reconhecimento de face, afim de alertar uma possível presença não permitida em um ambiente restrito, na realidade aumentada, sistemas são desenvolvidos para simular ambientes reais, simulando situações perigosas ou de difícil realização.



Figura 4. Segurança, Realidade Aumentada e Robótica, respectivamente.

3.4 Aprendizagem

Aprendizagem é uma propriedade das redes de extrema importância. A rede deve melhorar seu desempenho aprendendo com os estímulos de seu ambiente. Alguma métrica preestabelecida é utilizada para melhorar o desempenho e de maneira iterativa os pesos sinápticos da rede são ajustados para que a rede melhor responda aos padrões de entrada a cada nova iteração. Há diversos pontos de vistas e conceitos sobre aprendizado,

para o propósito de estudo e desenvolvimento das redes neurais artificiais, uma forma geral pode ser definido como em [20]:

“Aprendizagem é o processo pelo qual os parâmetros de uma rede neural são ajustados através de uma forma continuada de estímulo pelo ambiente no qual a rede está operando, sendo o tipo específico de aprendizagem realizada definido pela maneira particular como ocorrem os ajustes realizados nos parâmetros.”

Com essa definição podemos entender que o processo de aprendizagem ocorre da seguinte maneira [21]:

1. A rede recebe o estímulo do ambiente.
2. Como resultado desses estímulos a rede atualiza seus parâmetros livres.
3. Com as modificações na estrutura interna a rede vai responder de maneira nova aos próximos padrões de entrada.

Não existe nenhum algoritmo padrão para o treinamento das redes, existe um conjunto de técnicas que são usadas para gerar *algoritmos de aprendizagem*. O modo como a rede está estruturada internamente, a forma como seus neurônios estão interligados influenciam no modo de aprendizagem, porém a diferença entre os modos de aprendizado, é basicamente a forma pela qual são ajustados os sinápticos dos neurônios.

As formas como a rede pode aprender pode ser, entre outras: Aprendizagem por Correção de Erro, Aprendizagem Baseada em Memória, Aprendizagem Hebbiana, Aprendizagem Competitiva, Aprendizagem com um Professor e Aprendizagem sem um Professor. Para o desenvolvimento deste trabalho serão estudadas a Aprendizagem por correção de Erro, Aprendizagem com um Professor e Aprendizagem sem um Professor.

O treinamento supervisionado tem como característica o conhecimento prévia pela rede da resposta que se deseja obter como saída. Ou seja, além da entrada que a rede recebe para a realização do treinamento, a rede recebe também a saída desejada previamente. Com esses dados o treinamento é realizado de forma que a rede ajuste seus pesos sinápticos a fim de minimizar a diferença entre sua resposta e a resposta desejada.

É um treinamento caracterizado exatamente por ter no conjunto de treinamento, padrões que possuem um par entrada-saída. As entradas são os estímulos que são passados a rede para gerar sua resposta, e com essas saídas fazer a comparação com as saídas desejadas para obter o erro que será repassado para a correção dos pesos sinápticos.

No treinamento não-supervisionado a rede tem conhecimento apenas dos padrões de entrada, a rede não tem nenhum conhecimento prévio do que é esperado como resposta, como no treinamento supervisionado. A rede começa o treinamento sem saber qual a resposta de deve obter.

Apenas após algumas iterações é que a rede através das saídas geradas começa a obter dados que serão usados numa análise estatística para obter as informações sobre como ajustar melhor seus parâmetros para obter as saídas que melhor satisfaçam o problema.

Como na aprendizagem supervisionada existe a presença de um "professor", que apresenta a saída desejada, não há grande importância de como os padrões de treinamento sejam passados a rede. Entretanto no aprendizado não-supervisionado, só é possível a rede fazer uma análise estatística dos resultados, se houver redundância nos padrões de entrada, pois senão a análise feita pela rede não servirá para uma correta correção dos parâmetros.

3.4.1 Aprendizagem por Correção de Erro

Uma rede com um único nó de saída será usado para ilustrar o modo como ocorre a Aprendizagem por Correção de Erro. A rede recebe um vetor de entrada como estímulo do ambiente e a saída das camadas intermediárias são responsáveis por excitar o neurônio da camada de saída. A saída do neurônio da camada de saída é comparado com uma saída desejada que é conhecida. É calculado o erro e esse serve para a correção dos pesos sinápticos dos neurônios. O erro é mostrado na Equação 3 abaixo.

$$e_k(n) = d_k(n) - y_k(n)$$

Equação 3. erro entre a saída da rede e a saída desejada

Os erros calculados a cada iteração são representados por uma curva de erros. Dependendo do modelo da rede, se for composta de unidades lineares, apenas um mínimo local será encontrado, entretanto se houver unidades não-lineares, podem ser encontrados vários mínimos locais e um mínimo global. O objetivo do algoritmo de aprendizagem é percorrer essa curva e alcançar o mínimo global. Se o treinamento atingir um mínimo local, o treinamento não estará bom e deverá ser reiniciado. Uma regra muito utilizada para o ajuste dos pesos é a Regra Delta [4]. Na figura 5 abaixo são mostrados os mínimos locais e o mínimo global.

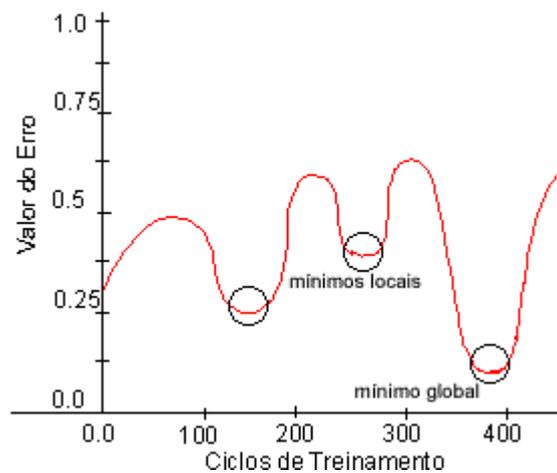


Figura 5. Uma possível superfície de erro com os mínimos locais e o mínimo global demonstrada graficamente

Com essa regra o peso sináptico tem seu ajuste dependente do erro calculado, do valor da entrada e da taxa de aprendizagem. Na iteração n para um estímulo k , observa-se a atualização dos pesos na Equação 4 abaixo.

$$\Delta w_{ij} = \eta e_k(n) x_j(n)$$

Equação 4. valor de ajuste dos pesos

Nos quais $\Delta w(n)$ – valor de ajuste a ser acrescentado ao peso w_{ij} ; η – taxa de aprendizado; $e(n)$ – valor do erro; $x_j(n)$ – valor do estímulo

Depois de atualizado o valor do peso será, na Equação 5:

$$w(n + 1) = w(n) + \Delta w_{kj}(n)$$

Equação 5. valor atualizado do peso

com a regra Delta corrigi-se os pesos sinápticos, e minimiza-se a função de erro, como na Equação 6:

$$\varepsilon(n) = \frac{1}{2}e^2(n)$$

Equação 6. função de custo

nos quais $e(n)$ – erro da rede na iteração n durante o treinamento;
 $\varepsilon(n)$ - valor da função de custo na iteração n durante o treinamento;

A taxa de aprendizagem é um parâmetro de grande importância no treinamento da rede, pois controla a alteração dos pesos. Uma taxa baixa deixa o aprendizado muito lento, já uma taxa alta, acelera o aprendizado mas pode causar muitas oscilações no treino e pode dificultar o processo de generalização da rede. Geralmente o valor da taxa varia de 0.1 a 1.0 [25].

4. RBF

As Redes RBF, redes com funções de base radial, são redes supervisionadas. As funções de ativação base radial são caracterizadas por ter uma resposta que cresce ou decresce de forma monotônica em relação a um ponto central. A RBF possui um treinamento híbrido, ou seja, esse tipo de rede neural tem seu aprendizado realizado das duas maneiras de treinamento. Primeiro é realizado o treinamento não-supervisionado e depois que esse treinamento é finalizado a rede tem uma nova etapa de aprendizagem que é o treinamento supervisionado. No projeto apresentado o algoritmo utilizado no treinamento não-supervisionado foi o k-médias e no treinamento supervisionado foi utilizado como forma de correção dos pesos sinápticos o algoritmo *backpropagation*. Nas próximas seções serão descritos com detalhes estes algoritmos.

As redes RBF são utilizadas em diversas aplicações, desde aplicações mais conhecidas como reconhecimento de caracteres, reconhecimento de voz até aplicações como diagnose médica, detecção de falhas, previsão e controle.

A figura 6 mostra a arquitetura de uma RBF com uma saída. Ela possui geralmente uma camada oculta.

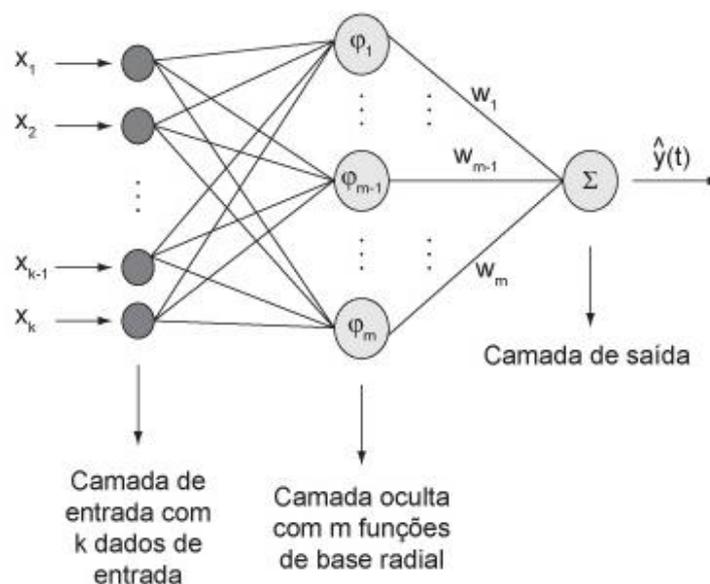


Figura 6. Rede RBF com uma saída

A rede recebe entradas não-lineares, e todos os padrões de cada entrada serve de estímulo para todos os nós da camada intermediária. Após o processamento da camada intermediária, a saída de cada nodo é ponderado por um peso e na camada de saída é feita uma operação linear para gerar a saída de resposta da rede.

No projeto da RBF é importante definir o melhor número de nodos da camada intermediária, pois os nodos agruparão os padrões em um *cluster*, esse padrões serão depois utilizados pela camada de saída.

A rede RBF possui como sua característica básica ser uma rede que tem como função de ativação da camada oculta uma função de base radial. Com essa função a RBF tem a ativação dos nós na camada intermediária dada pela distância entre seus vetores de entrada e de peso.

A camada intermediária de uma RBF tem seus nós definindo uma hiperelipsóide no espaço dos padrões de entrada [22]. A resposta da função de base radial está diretamente ligada com a distância dos padrões de entrada ao centro, ou seja, o valor será o maior a medida que os padrões estiverem perto do centro e menores quando estiverem bastante distantes, e o raio de influência já não estará influenciando tanto aquele padrão.

Com a característica de formar hiperelipsóides, as redes RBF só apresentam respostas às regiões do espaço de entrada que possuem dados de treinamento, pois são formados aproximadores locais. A figura 7 representa a partição dos dados para uma RBF de 3 nós, na camada intermediária.

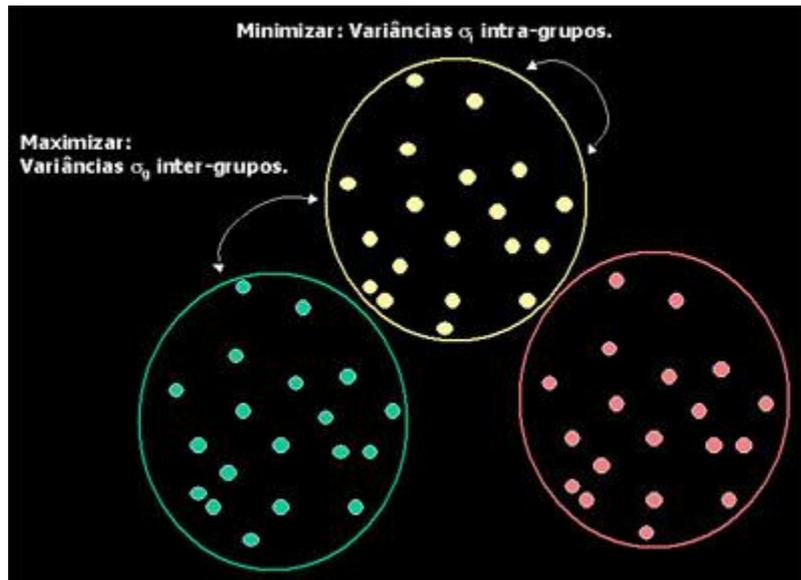


Figura 7. Partição de dados em clusters

4.1 Características da RBF comparadas com a MLP

Nesta seção será mostrada algumas características que demonstram o porquê do uso da RBF no projeto.

Tanto a MLP quanto a RBF são consideradas aproximadores universais, isso leva a conclusão de que sempre que pode-se usar uma MLP para resolver um problema pode-se usar uma RBF, e sempre que pode-se usar uma RBF há também uma MLP equivalente.

Algumas características são listadas abaixo fazendo a distinção entre a arquitetura e funcionamento das MLP e RBF[23]:

- Apesar de poder ter mais de uma camada intermediária as RBF's comumente possuem apenas uma, enquanto as MLP geralmente possuem mais de uma.
- Geralmente as camadas de uma MLP, seja de saída ou oculta, possuem a mesma função de ativação, já numa RBF os nós da camada intermediária e oculta, possuem funções muito diferentes e com objetivos diferentes para a rede.
- Geralmente num sistema de classificação de padrões as camadas de saída e ocultas de uma MLP são todas não-lineares. Já na RBF

a camada intermediária é não-linear, enquanto isso a camada de saída é linear.

- Nas RBF a função de ativação de cada nó da camada oculta calcula a distância euclidiana entre o vetor de entrada e o centro do nó. As MLP tem como cálculo da função de ativação de cada nó das camadas ocultas o produto interno entre o vetor de entrada pelo peso sináptico do nó.
- Enquanto a rede RBF constrói aproximadores locais, e por isso só oferece resposta as regiões do espaço que apresentam dados de treinamento, a MLP constrói aproximadores globais, que conseguem generalizar para regiões onde não existem dados de treinamento. Porém essa generalização gera o problema dos "falsos padrões", ou seja, a rede classifica nós completamente diferentes dos nós do treinamento como existentes.

4.2 Funcionamento da RBF no SARCO

Os padrões de entrada são apresentados um a um a rede, a primeira tarefa é realizar o treinamento da rede, que é dividido em não-supervisionado e supervisionado.

No treinamento não-supervisionado, são definidos os centróides da camada oculta e os raios de influência de cada centróide. A rede recebe um conjunto de padrões como estímulos, cada um desses padrões é processado pela camada intermediária. Os padrões de entrada são comparados com os centros de cada nodo da camada oculta através da distância euclidiana e associado ao nodo onde a resposta mostre maior semelhança. Esse procedimento é feito com todas as entrada, e refeito até que as associações entre cada padrão e os nodos não sofram modificações significativas. Com esse procedimento são definidos os centros. É preciso definir a região de influência de cada centro, ou seja, definir os raios de cada centro. O algoritmo utilizado para esse treinamento foi o k-médias, que será descrito em detalhes, junto com as formas adotadas para definição dos raios, na seção 4.4.1.

No treinamento supervisionado, os pesos sinápticos das camadas oculta e de saída são, a cada iteração, alterados até que o erro de treinamento não sofra alteração significativa. A rede corrige os pesos através do erro obtido na camada de saída. O erro é calculado como sendo a diferença entre a resposta obtida e a resposta que se deseja. O erro é retropropagado para corrigir os pesos sinápticos das camadas de saída e oculta. Quando não houver uma mudança importante nos ajustes dos pesos o treinamento é finalizado. Para realizar esse treinamento foi utilizado como base o *backpropagation*. O treinamento supervisionado e o *backpropagation* serão descritos nas seções 4.3 e 4.3.1 respectivamente.

Durante o treinamento é utilizado um conjunto de verificação utilizado para testar o quanto a rede consegue aprender a cada iteração. O processamento realizado por cada nó da camada oculta é feito através de funções de base radial e esse processamento é aplicado a todos os padrões de entrada. Serão descritos agora, em detalhes, os algoritmos utilizados na aprendizagem supervisionada e não-supervisionada, e as funções de ativação utilizadas na camada de saída e camada intermediária.

4.3 Aprendizagem Supervisionada

No treinamento supervisionado, a forma mais comum de treinamento de RNA, um agente externo fornece um par de entrada e saída desejada para a rede. O agente externo – professor – fornece a saída que representa um desempenho ótimo da rede. Um sinal contendo um erro é retornado a rede para que iterativamente a rede faça o ajuste necessário. Esse sinal é calculado como a diferença entre a saída desejada, fornecida pelo professor, e a saída real da rede, que é calculada com a entrada da rede e os pesos atuais. Dessa forma a rede vai aprender e adquirir o conhecimento que o professor detinha, e quando a rede conseguir classificar o mais corretamente possível os padrões de treinamento, e por consequência classificar os padrões de teste apresentados a rede não sendo mais necessária a presença do agente externo. A figura 8 abaixo representa o aprendizado supervisionado.

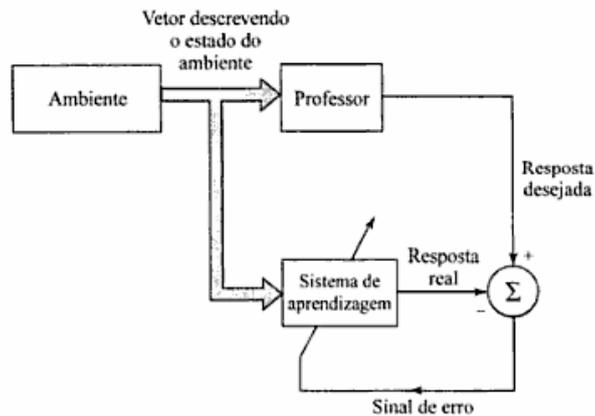


Figura 8. Esquema do treinamento supervisionado.

4.3.1 BackPropagation

O *backPropagation* é o algoritmo mais utilizado no treinamento supervisionado. Usado como forma de treinamento não só na RBF, mas em outros modelos de redes neurais como a MLP. O *backpropagation* é baseado na correção por erro, o erro é retropropagado da camada de saída para a camada oculta. Antes do uso do *backpropagation* não sabia-se como retropropagar o erro em RNA de múltiplas camadas. O *backpropagation* tornou-se um marco no campo das redes neurais, pois resolveu esse problema, que atrasou por muito tempo o uso de redes neurais, e porque provou ser um algoritmo suficiente no treinamento de redes[26].

O *backpropagation* é realizado basicamente com dois passos; o passo direto e o passo reverso.

No passo direto o padrão de entrada correspondente a excitação da rede é aplicada na camada de entrada e seu efeito se propaga de camada em camada até gerar uma saída na camada de saída. Esse passo caracteriza-se por possuir todos os pesos sinápticos fixos [26].

No passo reverso ocorre o ajuste dos pesos sinápticos. O ajuste é realizado com base no aprendizado por correção de erro. A saída da rede é comparada com a saída desejada e a diferença entre elas é o valor do erro que é retropropagado pelos mesmos neurônios anteriores, sendo que agora no sentido da camada de saída até a camada de entrada. Os pesos são

ajustados até a diferença entre as saídas desejada e obtida dada pela rede seja a mínima possível.

Será mostrado a seguir o algoritmo de treinamento supervisionado implementado no projeto SARCO. O algoritmo foi baseado no *backpropagation* e fez uso da Equação 7, Equação 8, Equação 9 abaixo:

$$erro_{absoluto} = |erro| = |(t_k - o_k)|$$

Equação 7. erro absoluto

$$\delta_k = o_k(1 - o_k) \times (erro)$$

Equação 8. variação da saída para ajuste dos pesos

$$\Delta w_j = \eta x_j \delta_j$$

Equação 9. valor para ajuste dos pesos

nas quais t_k – é a saída desejada; o_k – é a saída da rede; δ_k – é a variação da saída da rede; Δw_{ji} – representa o ajuste dos pesos das camadas ocultas; η – taxa de aprendizagem; δ_j – Variação da saída da rede; x_j – valor da unidade escondida.

A saída (o_k) da rede é calculada pela função sigmoideal que recebe como entrada a soma ponderada das unidades escondidas.

O valor do erro como já destacado é calculado fazendo-se a diferença entre a saída real da rede e a saída desejada previamente sabida, como mostrado na Equação 7 acima. O cálculo da saída e do valor do erro fazem parte do passo direto.

Já a Equação 8 e a Equação 9 apresentadas acima fazem parte do passo reverso, passo que retropropaga o erro e ajusta os pesos sinápticos da camada oculta.

As figuras 9 e 10 representam o passo direto e reverso respectivamente.

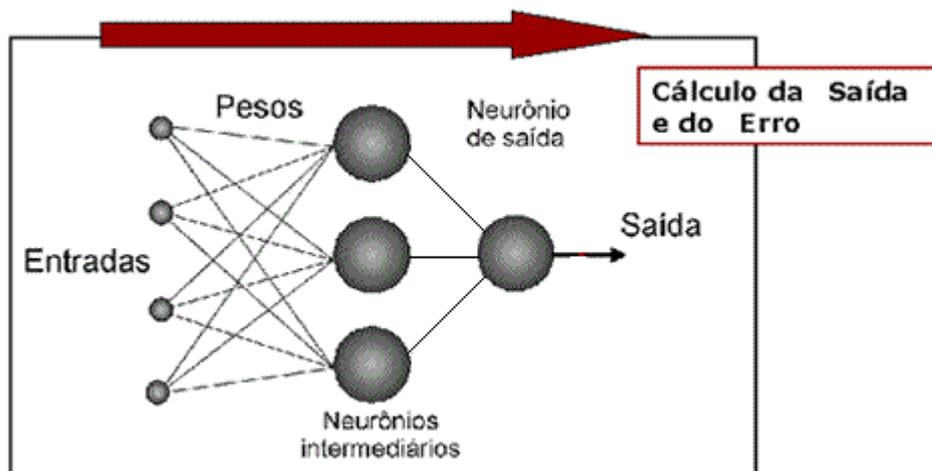


Figura 9. Passo direto do backpropagation.

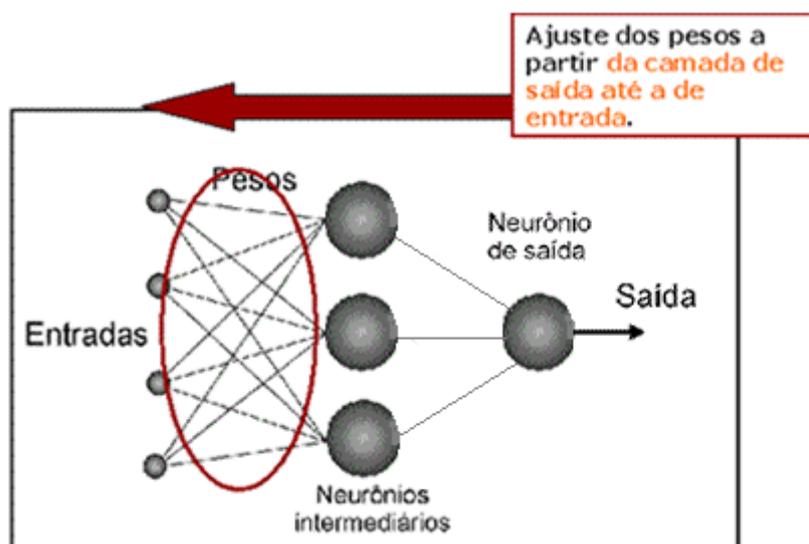


Figura 10. Passo reverso do backpropagation.

4.4 Aprendizagem Não-supervisionada

Chamada também de auto-organizada [21], não possui um crítico ou professor informando a saída desejada. Ao invés disso, a rede recebe dados de uma medida estatística, e a partir das regularidades estatísticas obtidas pela entrada dos dados, as redes tornam-se capazes de criar novas classes automaticamente. Para que isso possa acontecer é necessária a redundância nos dados de entrada, pois a rede irá extrair padrões desses dados, algo que sem redundância não será possível ocorrer. Existem alguns algoritmos de aprendizagem não-supervisionada, tais como,

aprendizado hebbiano, modelo de *Linsker*, aprendizado por reforço. A figura 11 abaixo mostra o aprendizado não-supervisionado.

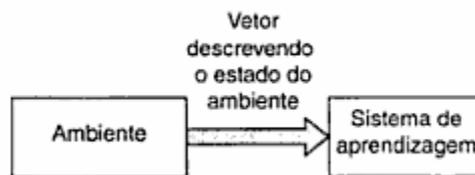


Figura 11. Esquema do treinamento não supervisionado.

4.4.1 Algoritmo do *k-médias* e seu uso no Sarco

O algoritmo *k-clustering* foi desenvolvido por J. MacQueen em 1967. O *k-médias* é um algoritmo de agrupamento não-hierárquico, nele os dados são particionados e métodos estatísticos são usados para classificação de padrões. O número de classes do *k-médias* deve ser definido a priori e os centróides devem ser iniciados aleatoriamente, isso caracteriza o fato do algoritmo ser não-determinístico.

O algoritmo *k-médias* tem como objetivo separar os dados de uma base em *k* classes, de forma que em cada classe sejam observadas similaridades entre os dados. Os centróides são inicializados aleatoriamente, porém uma inicialização em pontos regulares e com uma distância igual entre eles agilizam o processo de agrupamento [17].

Para fazer a definição dos centróides são realizados dois cálculos. No primeiro fazemos a distância Euclidiana, equação 4.4, entre cada uma das amostras e os centróides afim de identificar a qual centróides elas pertencem; as amostras são associadas aos centróides que se encontram mais perto. O cálculo da distância Euclidiana é feito comparando-se cada dimensão do ponto amostrado com o ponto correspondente de cada centróide, e está representada na Equação 10 abaixo.

$$d(n, m) = \sqrt{\sum_{i=1}^i (x_{in} - x_{im})^2}$$

Equação 10. distância euclidiana

No qual *n* e *m* representam os pontos.

Após todos os pontos terem determinados seus centróides correspondentes é feita uma média, que pode ser uma média geométrica ou aritmética, para que o centróide seja reposicionado e esteja agora no ponto central em relação a todos os que estão agrupados a ele, diminuindo assim a diferença entre ele e os pontos a ele associado.

Com os centróides reposicionados devem ser calculadas novamente as distâncias dos pontos aos centróides fazendo novas associações entre os pontos e os centróides, aí são refeitas as médias e o reposicionamento dos centróides, esses passos devem ser seguidos até que não haja mais mudança no posicionamento dos centróides. A média aritmética é representada abaixo, na Equação 11.

$$\bar{x}_i = \frac{1}{N_i} \sum_{x \in X'} x$$

Equação 11. média aritmética

no qual x – cada uma das amostras; N – o número total de amostras.

O algoritmo do k-médias [8] é definido como:

1. Supondo inicialmente o k-médias ($k=3$), são aleatoriamente selecionados no conjunto de dados.
2. Os k *clusters* são criados associando para cada observação a média mais próxima. As partições geradas são representadas pelo Diagrama de Voronoi.
3. Os centróides de cada um desses k -*clusters* tornam-se os novos centros.
4. Os passos 2 e 3 são repetidos até a convergência ter sido alcançada.

Os passos de 1 a 4 estão demonstrados nas imagens abaixo:

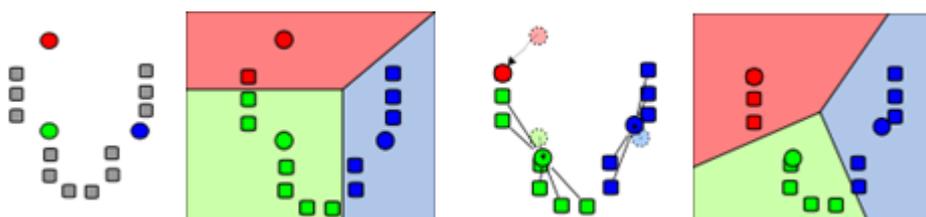


Figura 12. Ilustração do k-médias.

O algoritmo explicado acima foi desenvolvido no projeto como algoritmo do treinamento não supervisionado, afim de obter os agrupamentos corretos para classificação correta dos ovos e fundos das imagens.

Após a definição de todos os centros, é ainda realizado no treinamento não-supervisionado, a determinação dos raios, ou seja, a área de influência de cada um dos centróides.

No Projeto desenvolvido os raios foram determinados com duas abordagens distintas. Na primeira abordagem adotada, cada centro possuía um raio, ou seja, sua região de influência. Esse cálculo era realizado pela distância euclidiana entre cada centro e todos os outros e a menor distância obtida era associada como raio daquele centro. Para o segundo conjunto de dados de treinamento foi calculada a distância euclidiana entre todos os centros e a maior distância associada como os raios de todos os centros. Os conjuntos de treinamento utilizados na primeira e segunda abordagens serão descritos no capítulo 5.

Será apresentado o pseudo-código do k-médias utilizado no projeto[20] e depois como foram definidos os raios:

1. Inicialize os centros com valores aleatórios;
2. Associar os primeiros K vetores de entrada aos K centros;
3. Associar cada novo vetor de entrada ao grupo mais próximo;
4. Com todos vetores agrupados, calcular novamente os centros;
5. Repetir até que não haja mais mudança nos centros:
 - 5.1 Para cada vetor de entrada:
 - 5.1.1 Calcular a distância para todos os centros;
 - 5.1.2 Se o vetor não pertencer ao grupo mais próximo
Associe o vetor ao centro mais próximo

Recalcule os centros dos grupos dois grupos, o que perdeu e o que ganhou o vetor

Os raios, na primeira abordagem:

1. Para cada centro calcule a distância euclidiana entre ele e todos os outros centros.
2. Associe como raio do centro a menor distância obtida para cada centro no passo 1.

Os raios, na segunda abordagem:

1. Calcule a distância euclidiana entre todos os centros.
2. Associe a maior distância obtida no passo 1 como raio de todos os centros.

No projeto realizado o algoritmo do k-clustering dividia o conjunto em duas classes, uma representava os ovos e a outra fazia a representação dos fundos das imagens.

4.5 Função de Ativação da Camada de Saída

No projeto do SARCO foi utilizada a função Sigmóide como função de ativação da camada de saída. A saída da função assume valores no intervalo [0 1]. A forma mais utilizada da função Sigmóide está representada na Equação 12 representada abaixo. A figura 13 abaixo mostra o gráfico da curva sigmóide

$$f(v) = \frac{1}{1 + e^{(-av)}}$$

Equação 12. função sigmóide

no qual, a – é o parâmetro de inclinação da função sigmóide; v – é o valor de ativação do neurônio

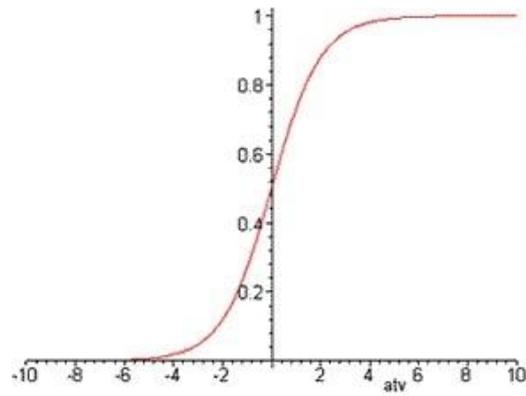


Figura 13. Gráfico da Função Sigmóide

no qual v é dado na Equação 13:

$$v(i) = \sum_{i=1}^n w_i y_i$$

Equação 13. soma ponderada das saidas

no qual y_i são as saídas de cada nodo da camada oculta; w_i são os pesos correspondentes a cada nó da camada oculta.

5. Proposta do SARCO

O projeto SARCO - Sistema Automático de Reconhecimento e Contagem de Ovos da dengue - tem a proposta de auxiliar a contagem dos ovos de maneira automática. Para a realização desse trabalho foi buscado um mecanismo inteligente de contagem baseado em redes neurais. O projeto visa desenvolver um sistema de rede neural, no qual, de posse das imagens das ovitrampas seja possível de forma totalmente automática, a contagem dos ovos existentes em cada uma das ovitrampas. O sistema desenvolvido será feito de forma que uma vez treinado com todas as informações necessárias, uma vez que o sistema tenha aprendido o que o que são ovos e o que são fundos, ele faça o reconhecimento e contagem dos ovos em todas as palhetas que forem recolhidas do meio ambiente. Abaixo na Figura 14. Estrutura do SARCO, com seus diversos módulos de processamento. será mostrada a arquitetura geral do SARCO.

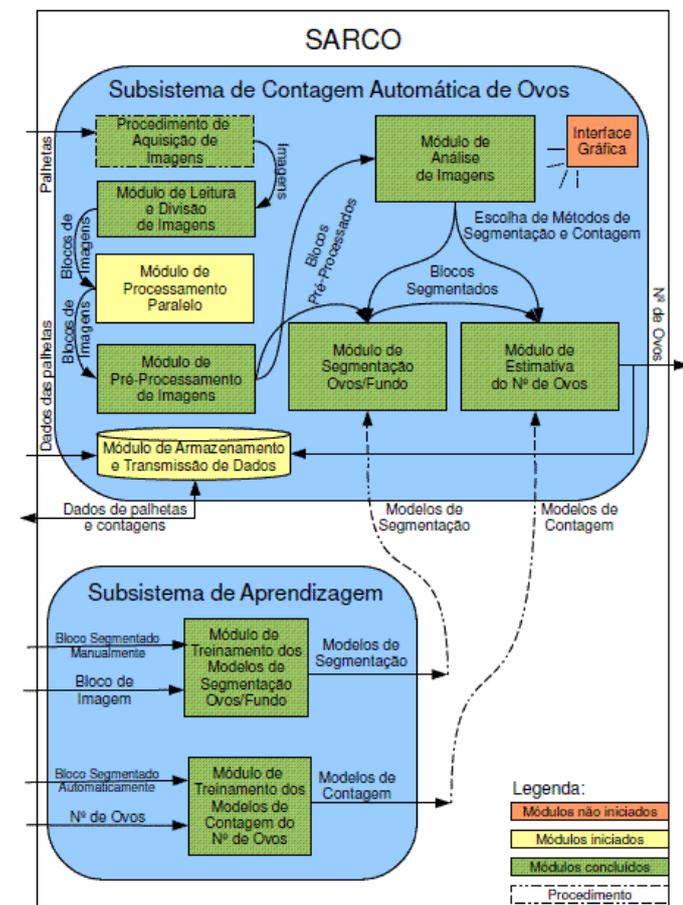


Figura 14. Estrutura do SARCO, com seus diversos módulos de processamento.

A rede RBF está incluída nos Módulos de Análise de Imagens, Módulo de segmentação Ovos/Fundo e Módulo de Treinamento dos Modelos de Segmentação Ovos/Fundo.

O sistema SARCO é desenvolvido com objetivo de ser um sistema automático útil ao usuário e de fácil manuseio. A rede RBF será responsável no sistema, por fazer de forma autônoma o reconhecimento e contagem dos ovos. O usuário fornecerá ao sistema os dados para análise, esses dados serão as entrada para a rede. Esses dados são extraídos das imagens das palhetas digitalizadas e serão analisados pelo sistema fazendo uso da rede com objetivo de informar a quantidade de ovos que cada palheta contém.

A análise dos resultados de teste, levou a uma alteração do algoritmo de treinamento supervisionado. O treinamento antes feito de forma igual tanto para fundos como ovos, foi alterado para treinar de forma diferente um padrão que representasse um ovo de um padrão que representasse um fundo.

Como nas imagens das ovitrampas é possível observar que a quantidade de fundo é muito maior que a de ovo, ou seja, não há uma proporção de igualdade, é importante treinar a rede de forma que ela possa reconhecer muito bem os padrões de fundo. Isso é necessário porque se a rede classificar erradamente os padrões de fundo, como a imagem tem uma enorme quantidade de fundo, isso gerará uma classificação errada no número de ovos, o que fará com que a rede gere resultados ruins para a classificação no número de ovos. As palhetas contém uma grande quantidade de sujeiras e manchas, o que leva ao aparecimento de muitos ruídos quando são extraídas as suas imagens. Foi observado que esses ruídos possuem uma tonalidade de cor muito semelhante aos ovos, levando a uma dificuldade no reconhecimento e contagem dos ovos, pois a rede não consegue fazer a distinção entre esses ruídos e os ovos. Fazendo a associação errada dos ruídos aos ovos, encontrou-se muitas dificuldades em se obter valores satisfatórios na contagem das palhetas que continham muitos ruídos.

Fazendo essa análise da proporção da quantidade de ovos e fundo a Equação 8 foi alterada. A rede verifica qual o padrão de entrada está

recebendo, caso seja ovo, os pesos são alterados de uma forma mais suavizada, caso seja um padrão de fundo os ajustes ocorrem de forma mais acentuada. Essa alteração é feita através de fator ponderante incluído na Equação 8 e que tem como função diferenciar o modo como a rede ajusta os seus pesos para os padrões de ovo e de fundo. O fator β deve estar numa faixa de valores entre $0.1 < \beta < 1$ para o ajuste dos pesos dos ovos e com valor $\beta = 1$ para ajustar os pesos dos fundos. Não foi adicionado valores de $\beta > 1$ para ajustar os pesos dos fundos pois estes valores geraram estabilidade na rede, essa estabilidade foi detectada através de testes realizados. Para a faixa de valor utilizada a rede aprenderá mais precisamente o que é fundo em relação ao que é ovo. Dessa forma ao testar as imagens, as repostas dadas pela rede estarão com uma precisão maior na qualificação do fundo, o que é bastante interessante para a resposta final da rede visto a grande diferença proporcional entre ovos e fundos numa palheta. Ou seja, a rede dará como valor final para contagem dos ovos, valores mais pertos dos reais.

5.1 Função de Base Radial

A função de base radial utilizada na camada intermediária foi uma gaussiana. Nela temos como parâmetros a distância entre os padrões e os centros e os raios. Abaixo, na Equação 14 temos a função gaussiana utilizada como base para a implementação.

$$f(x) = e^{(-r * (x-x_0)^2)}$$

Equação 14. função de base radial

na qual x – é o padrão de entrada; x_0 – é o centro; r – é o raio.

5.2 Aquisição das Imagens

A aquisição das imagens consiste na metodologia empregada para digitalização das palhetas em imagens.

A opção de digitalização adotada foi o *scanner* HP G4050, que tem como características o fácil manuseio e grande velocidade de digitalização, podendo também digitalizar várias palhetas simultaneamente, dependendo do tamanho, 5 ou 6 palhetas. Possui um sistema de iluminação uniforme e homogeneia que é bastante importante para não gerar regiões mais escuras que outras na digitalização. A sobreposição de imagens é evitada pelo fato das palhetas poderem ser digitalizadas por completa em uma única imagem. Como grandes desvantagens pode-se citar o tempo de aquisição demorado, dependendo da resolução escolhida, e a impossibilidade de acoplamento de um sistema de iluminação que possibilitasse a geração de imagens com maior qualidade.

Cerca de 300 palhetas foram digitalizadas com esse *scanner*. A taxa média de digitalização foi de 5 palhetas/hora para resolução de 2400dpi, e de 10 palhetas/hora para resolução de 1200dpi. O tamanho médio das imagens foi de 71MB para digitalização em 2400dpi, e de 21MB para as mesmas digitalizações em 1200dpi. As palhetas também foram digitalizadas nesta resolução mais baixa (1200dpi) porque não foi descartada a possibilidade do SARCO trabalhar em uma resolução mais baixa que a definida como padrão(2400dpi).

Os procedimentos para aquisição das imagens são de fundamental importância para uma contagem precisa. Dois problemas relevantes, relativos à aquisição das imagens foram encontrados ainda durante os testes de contagem realizados com o SCAC: sujeira e/ou manchas na palheta e nível significativo de embassamento no foco da imagem adquirida. Esses problemas detectados no SCAC deverão ser ainda mais perceptíveis no SARCO, visto que como um sistema automático é ainda mais sensível aos problemas citados

Existem armadilhas que ficam armadas por longos períodos de tempo (de 3 a 4 semanas) com o único objetivo coletar o máximo de ovos e impedir sua eclosão. Suas palhetas são em geral, muito sujas e escuras (mudam do marrom para o preto ou verde). Existem outras armadilhas que são recolhidas semanalmente ou a cada 15 dias e estas sim visam avaliar o grau de infestação do *Aedes aegypti* através da contagem de seus ovos. O

fato de ficarem menos tempo expostas ao ambiente faz também, que em média, sua coloração não esteja tão escura e a quantidade de sujeira seja bem menor.

O outro grande problema encontrado foram as imagens desfocadas de palhetas. Isso ocorreu basicamente porque várias palhetas estavam empenadas ou retorcidas. Isso deve ser resultado do processo de transporte e armazenamento das mesmas após o recolhimento das armadilhas, visto que as palhetas são feitas de um material rígido. Elas amolecem e são contorcidas após ficarem imersas em soluções aquosas por um longo período de tempo. A perda da forma plana das palhetas acarreta que partes de imagens de uma mesma placa necessitem de focos distintos, fato que não ocorre, levando a produzir imagens sem boa nitidez.

Portanto, para a implementação do SARCO (que tende a ser ainda mais sensível a esses tipos de problemas), como mecanismo de contagem em grande escala, faz-se necessário refazer o planejamento tanto do tempo de coleta das armadilhas quanto do seu transporte e armazenamento. Essas etapas serão muito importantes para a precisão do sistema de contagem escolhido.

Os dados para a calibração do SARCO foram obtidos através da realização de testes de contagem com o SCAC. Uma vez que são realizados testes com o SARCO, é necessária uma avaliação do quão bons foram seus resultados. Para isso são necessários conjuntos de dados para fazer a comparação entre os resultados e avaliar os resultados do SARCO, esses conjuntos são os conjuntos dos testes de contagem realizados pelo SCAC. É necessário saber se foram geradas boas estimativas de contagens para palhetas, tenham elas muitos ou poucos ovos.

5.3 Extração de Características das Imagens das Palhetas

O sistema desenvolvido recebe como entradas arquivos de texto. Esses arquivos são conjuntos de treinamento e de validação preparados para realizar o treinamento da rede e ainda um conjunto de teste. Os conjuntos

de treinamento de validação são compostos por dados extraídos da imagem, porém nesses dados é necessário que haja a presença de informações sobre ruídos, fundos e ovos, para que a rede possa aprender a definir cada um. O conjunto de teste possui os dados das imagens a serem testadas após o treinamento da rede. A seguir será descrito o processo de extração das características.

Com a ajuda da Figura 15, pode-se entender a composição dos atributos de cada conjunto de descritores. O Conjunto de Descritores 1 (CD1), foi criado a partir da máscara quadrada 3x3 vermelha. O pixel da imagem analisado, é aquele no centro da máscara quadrada sob o pixel em amarelo. O pixel em questão está descrito por 27 atributos. Esses atributos são os valores R, G, B dos pixels da imagem que estão sob a máscara quadrada 3x3 (incluindo as componentes RGB, do próprio pixel analisado).

O outro conjunto de descritores, chamado de Conjunto de Descritores 2 (CD2), é composto por dados gerados a partir de duas máscaras (Figura 15). O pixel analisado é mesmo citado no parágrafo anterior. Sua descrição consiste de 9 atributos: as médias das componentes R, G, B, dos pixels das imagens que estão sob a máscara central vermelha de tamanho 3x3; as médias R, G, B, e seus respectivos desvios padrões σ_R , σ_G , σ_B , do conjunto de pixels da imagem que estão sob a máscara circular vermelha.

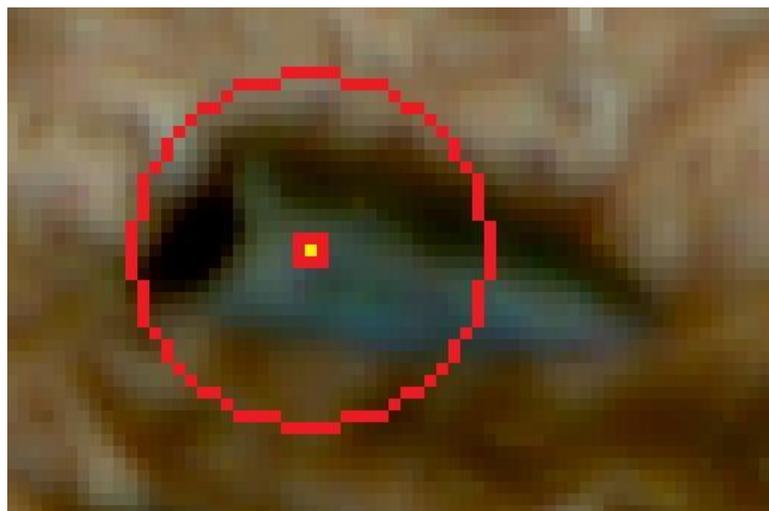


Figura 15 - Imagem de um ovos e um conjunto de máscaras quadrada e circular (em vermelho) utilizadas para descrever um pixel da imagem (que está sob o ponto amarelo no centro da máscara vermelha quadrada).

Um conjunto de padrões de treinamento, formado por padrões de OVOS e de FUNDO na mesma proporção, foi construído a partir de amostragem aleatória em 50 imagens contendo as mais diferentes tonalidades de ovos e em 30 imagens ruidosas (imagens que não possuem ovos e não possuem a tonalidade marrom típica da palheta). A Figura 16 trás alguns exemplos dessas imagens.

As imagens da linha central são as mesmas imagens da linha superior, porém segmentadas manualmente. Nesta rotulação *a priori*, os pixels em preto dentro da região cinza são considerados pixels da classe OVO. Os pixels da região cinza são classificados com INDEFINIDOS por não se saber exatamente a borda da imagem do ovo. O restante da imagem é rotulado como sendo da classe FUNDO.

Através da segmentação *a priori* foi possível se construir conjuntos de dados que contenham as duas classes e que foram utilizados para treinar os classificadores. Os pixels rotulados como INDEFINIDOS foram impedidos de compor o conjunto de treinamento, uma vez que sua inclusão traria mais ruído ainda para o problema.



Figura 16 - Imagens da 1ª linha (com ovos) e 3ª linha (sem ovos, apenas ruído) foram utilizadas para criar o conjunto de treinamento. Imagens centrais são as segmentações manuais das imagens da 1ª linha, ou seja, são as rotulações de todos os pixels das imagens da 1ª linha.

6. Problemas na Implementação, Solução para os Problemas, Testes e Resultados

6.1 Resultados

Antes de demonstrados os resultados será feita uma descrição dos conjuntos de treinamento, validação e configuração da rede.

Os resultados apresentados nesta seção, contemplam os testes realizados sobre as nove palhetas descritas na seção anterior.

- Redes treinadas e validadas com os dados gerados pelos descritores CD1, CD2 e CD2 reprocessado ($CD2_{repr}$).
 - Conjunto C1: 9000 padrões para treinamento (4500 de cada classe); 3000 padrões para validação (9/10 de padrões da classe FUNDO e 1/10 de padrões da classe OVO). Todos gerados com o conjunto de descritores CD1.
 - Conjunto C2: mesma configuração do C1, gerado com o conjunto de descritores CD2.
 - Conjunto $C2_{repr}$: 6000 padrões para treinamento (3000 de cada classe); 3000 padrões para validação (9/10 de padrões da classe FUNDO e 1/10 de padrões da classe OVO), com o conjunto de descritores $CD2_{repr}$.
- Os descritores do $CD2_{repr}$ são os mesmos descritores do CD2, porém eles foram aplicados às imagens processadas pelas redes (estas imagens possuem apenas o tom-de-cinza). Dessa forma, o conjunto de atributos se reduz a 3: média da máscara 3x3, média e desvio padrão da máscara circular.
- Todos os blocos das nove palhetas foram testados com redes treinadas e validadas com os conjuntos de dados C1, C2 e $C2_{repr}$.
- Após o treinamento das Redes Neurais, durante os testes realizados na etapa de separação OVO-FUNDO, percebeu-se a necessidade de uma seleção das palhetas que poderiam ser analisadas.

Duas configurações básicas foram adotadas para RBF, depois que testes preliminares mostrarem bons resultados para uma mesma configuração de rede, quando executadas com conjunto de dados diferentes.

A primeira configuração escolhida e utilizada para o conjunto de treino C1:

- 40 neurônios na camada escondida completamente conectados com os dados de entrada e com um neurônio na camada de saída.
- A saída da rede foi transformada para ficar no intervalo $[0,1]$, o que representa a probabilidade do pixel pertencer a classe FUNDO.
- Neurônios com função de ativação do tipo sigmóide.
- Taxa de aprendizagem de 0.32 e viés 0.2
- Dois critérios de parada foram utilizados: o número de iterações ($\# < 3000$) ou a perda de generalização pelo crescimento do erro do conjunto de Validação. Se o erro crescesse por mais de 5 épocas seguidas, em relação a época $t-5$, então parava-se o treinamento.

A segunda configuração escolhida e utilizada para os conjuntos de treino C2 e $C2_{repr}$:

- 20 neurônios na camada escondida completamente conectados com os dados de entrada e com um neurônio na camada de saída.
- A saída da rede foi transformada para ficar no intervalo $[0,1]$, o que representa a probabilidade do pixel pertencer a classe FUNDO.
- Neurônios com função de ativação do tipo sigmóide.
- Taxa de aprendizagem de 0.16 e viés 0.4
- Dois critérios de parada foram utilizados: o número de iterações ($\# < 3000$) ou a perda de generalização pelo crescimento do erro do conjunto de Validação. Se o erro crescesse por mais de 25 épocas seguidas, em relação a época $t-25$, então parava-se o treinamento.

A Tabela 1 trás os resultados das simulações com a RBF para os três conjuntos de dados testados. Para o conjunto C2 reprocessado por exemplo, as redes conseguiram classificar corretamente 90% dos padrões de FUNDO do conjunto de treinamento, e 92.7% dos padrões de FUNDO do conjunto de validação.

As Redes Neurais foram ponderadas, com o fator β , para forçar um aprendizado mais eficiente da classe FUNDO, como mostram as taxas de classificação correta da Tabela 1. Uma vez que o número de pixels da classe FUNDO é imensamente maior que o da classe OVO, é fundamental aumentar a precisão para classificar corretamente os pixels do fundo da imagem.

		C1	C2	C2 _{repr}
Taxa de Classificação Correta (Treino)	OVO	90%	90.5%	91%
	FUNDO	89%	92.6%	90%
	LIMIAR	0.476	0.06	0.125
Taxa de Classificação Correta (Validação)	OVO	70%	80%	87%
	FUNDO	96%	96.7%	92.7%
	LIMIAR	0.070	0.0078	0.0078

Tabela 1 - Resultados das simulações com Rede Neural RBF para os dados de treinamento e validação, para os três conjuntos de dados CD1, CD2 e CD2_{repr}

Devido a diferença entre o conjunto de validação e treinamento é possível observar que as redes RBF estão bem treinadas, dada que as taxas de acerto do conjunto de validação mostram a generalização do aprendizado.

O primeiro exemplo ilustrativo da segmentação de uma imagem através de uma rede RBF pode ser visto na Figura 17. Esta figura trás uma imagem de um bloco, a estimativa da segmentação ótima desejável, ou seja, segmentação manual, e os resultados de segmentação da RBF com um limiar de binarização de 0,5 para os conjuntos C1, C2 e C2_{repr}.

Observa-se que a segmentação para o Conjunto C1, apresenta-se menos ruidosa em comparação aos outros dois conjuntos. Existem várias pequenas regiões de pixel de ovos, que são na verdade, ruído que a RBF não foi capaz de retirar. Em todas as segmentações, as redes foram capazes de aprender regiões com e sem sobreposição de ovos, embora nem todos os ovos foram completamente reconstruídos. A rede RBF

reprocessada com o Conjunto C2, obteve diferente do esperado, uma segmentação mais "suja". O ruído está presente em maior quantidade.

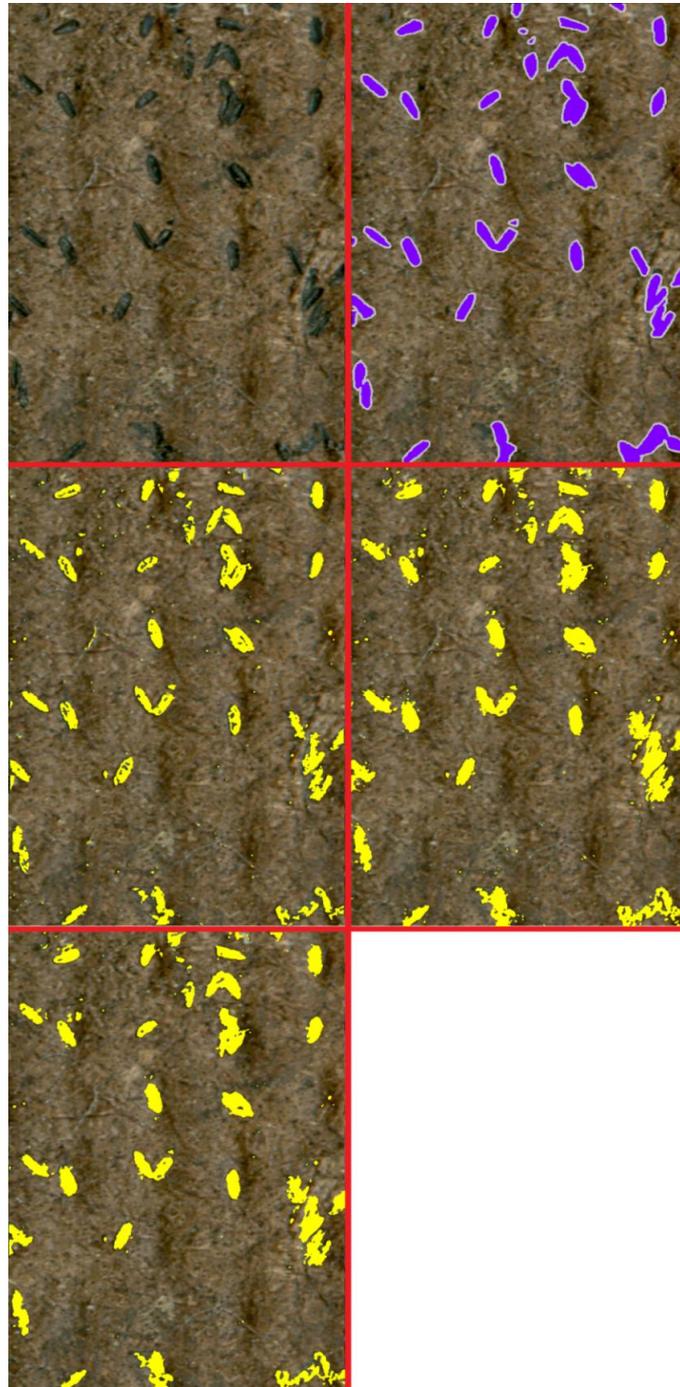


Figura 17 - esquerda superior: imagem de um bloco; direita superior: segmentação manual; esquerda central: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS em amarelo); direita central: segmentação com RBF treinada com o Conjunto C2; esquerda inferior: segmentação com RBF treinada com o Conjunto C2_{repr}.

O segundo exemplo de segmentação de blocos de imagens utilizando a rede RBF, está na Figura 18. Neste bloco não há ovos, porém as três segmentações apresentam várias regiões com pixel de ovos. Este tipo de problema foi muito comum nas simulações. Pode-se observar vários pontos pretos em toda a placa. Além disso, a própria estrutura da palheta, que possui pequenas "covas", se confunde com ovos, devido a aquisição das imagens (estas regiões não conseguem ser bem iluminadas pelo *scanner* na hora da digitalização da imagem, ficam sombreadadas, parecendo assim com a tonalidade de ovos). Vale-se salientar que estas segmentações foram classificadas como válidas, ou seja, este bloco não foi excluído nem na Seleção₁, nem na Seleção₂.

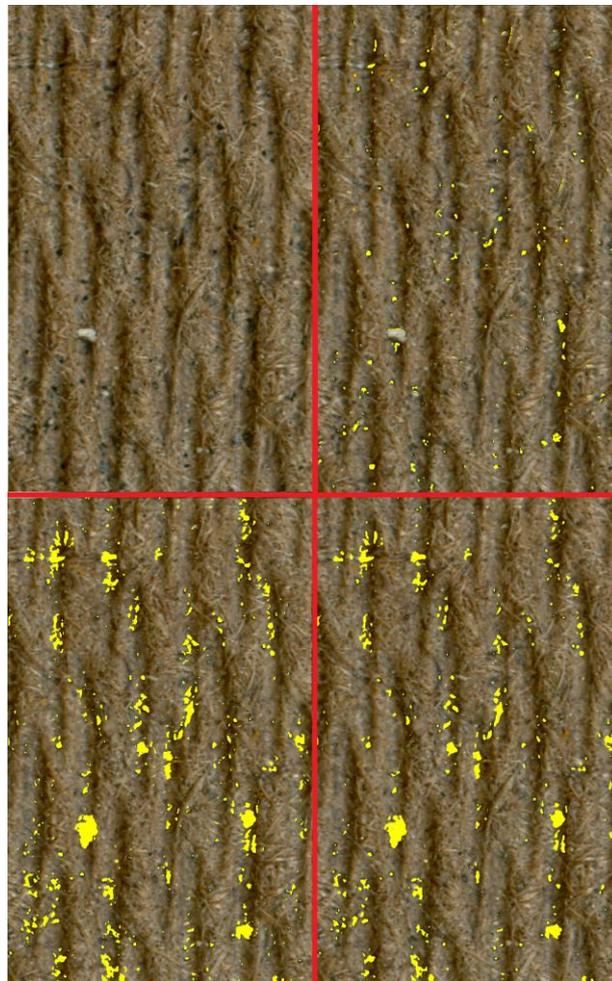


Figura 18 – esquerda superior: bloco sem ovos e com manchas; direita superior: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS pintados em amarelo); esquerda inferior – segmentação com RBF treinada com o Conjunto C2; direita inferior – segmentação com RBF treinada com o Conjunto C2_{repr}.

Um exemplo de bloco, excluído pela Seleção₁ pode ser visto na Figura 19. Em grande parte das imagens, os pixels das bordas da palheta acabam sendo detectadas como pixels de ovo, quando na verdade são ruídos. A segmentação com o conjunto C2 reprocessado apresentou melhora nas bordas, suavizando-as, mas a imagem toda ficou mais ruidosa do que a obtida pela segmentação com o conjunto C1.

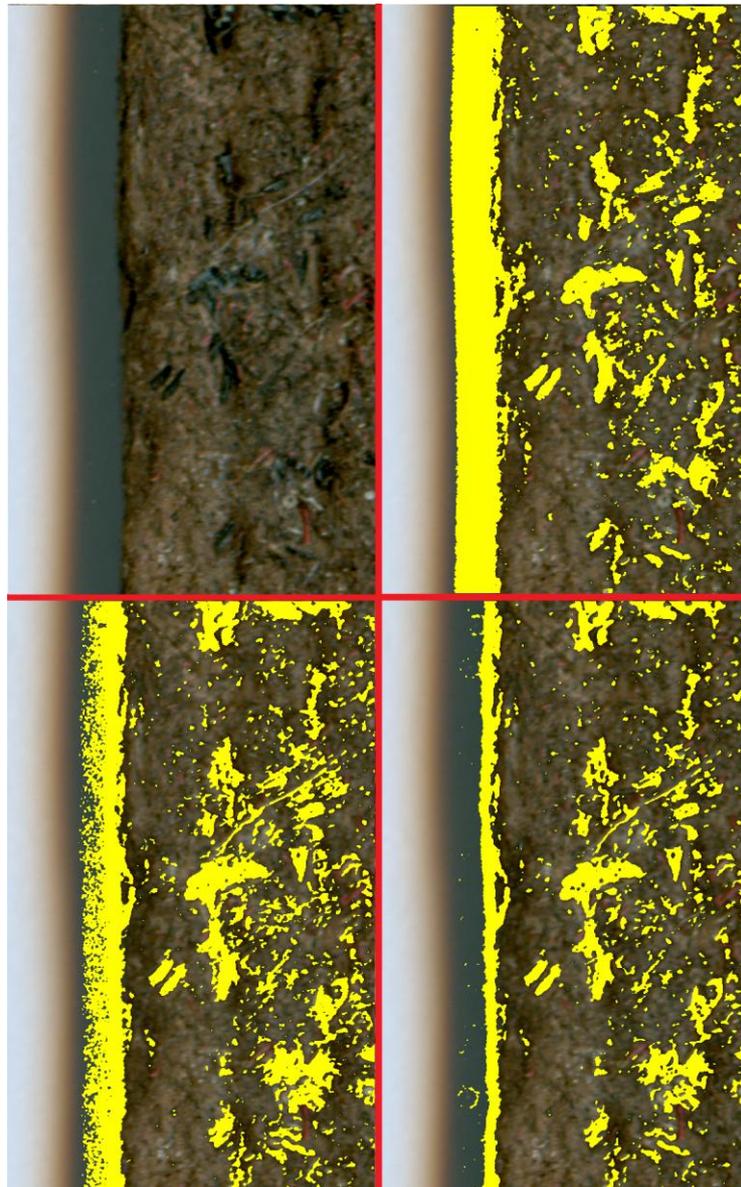


Figura 19 - esquerda superior: bloco da borda da palheta, excluída na Seleção₁ de blocos; direita superior: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS em amarelo); esquerda inferior: segmentação com RBF treinada com o Conjunto C2; direita inferior: segmentação com RBF treinada com o Conjunto C2_{repr}.

Uma segunda seleção de blocos foi estabelecida para retirar à mão, blocos cuja segmentação resultou em grande quantidade de pixel de ovos (quando na verdade havia mais sujeira que ovos no bloco). A Figura 20 mostra um exemplo típico desse problema.

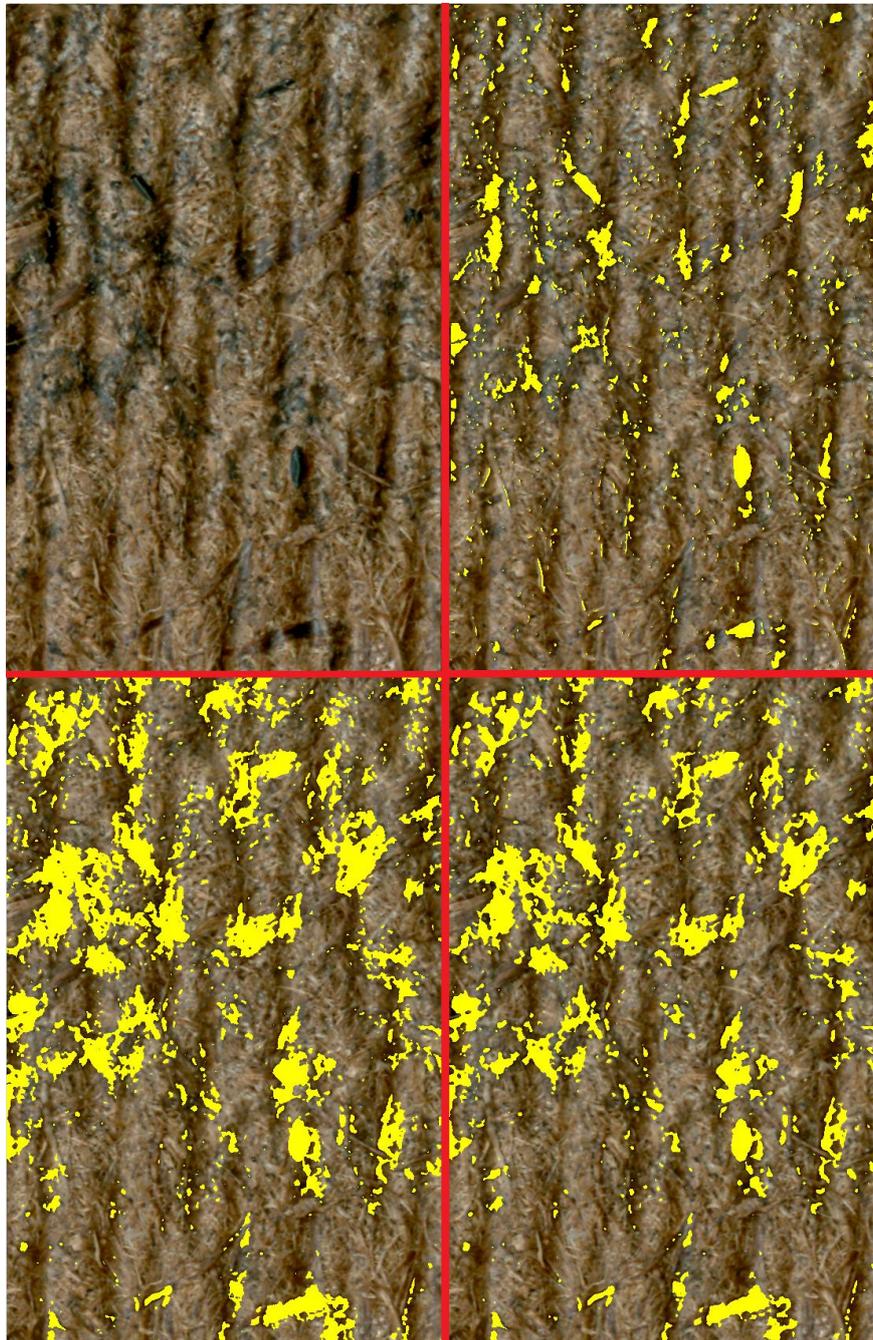


Figura 20 - esquerda superior: bloco com mancha, excluído na Seleção₂ de blocos; direita superior: segmentação com RBF treinada com o Conjunto C1 (pixels classificados como OVOS em amarelo); esquerda inferior: segmentação com RBF treinada com o Conjunto C2; direita inferior: segmentação com RBF treinada com o Conjunto C2_{repr}.

6.2.1 Estimativa do Número de Ovos

A estimativa do número de ovos em uma palheta, foi realizada a partir da estimativa do número de ovos por bloco. Uma vez otimizado essa contagem, o número de ovos da palheta será a soma do número de ovos de todos os blocos da palheta.

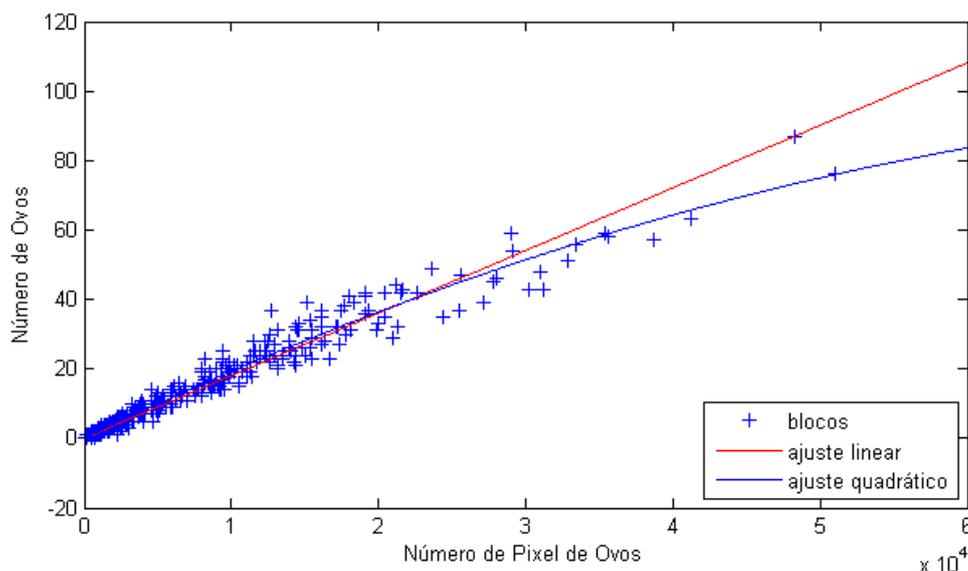


Figura 21 - Ajustes linear e quadrático para os dados de segmentação manual de 9 palhetas (998 blocos).

Inicialmente, começou-se a trabalhar com os dados referentes à segmentação manual na procura de uma função que ajustasse estes dados, que são os valores mais próximos dos reais. Percebeu-se que o uma função linear ou quadrática, poderia servir bem com função de ajuste. A Figura 21 trás os melhores ajustes linear(vermelho) e quadrático(azul) para os blocos de 9 palhetas, que representam 998 blocos segmentados manualmente. Os ajustes realizados nesta seção possuem 95% de confiança na estimativa de todos os seus coeficientes.

Seja o erro quadrático médio do ajuste definido como, na Equação 15 :

$$erro = \sqrt{\frac{\sum_{i=1}^n (NO_i - ajuste_i)^2}{n-1}}$$

Equação 15. erro quadrático médio

No qual NO_i – o número de ovos contados; $ajuste_i$ – o número de ovos estimado pelo ajuste no i -ésimo bloco.

Os valores de erro médio quadrático para os ajustes linear e quadrático, para as três seleções de blocos adotadas estão na Tabela 2. No melhor caso, ao se estimar o número de ovos em um bloco, erra-se em média por 1,75 ovos. Para 95% dos blocos, esse erro é de no máximo 3,5 ovos. No pior caso, o erro médio de ovos por bloco é de 2,37 ovos, e para 95% dos blocos, erra-se no máximo, 4,74 ovos por bloco.

	Ajuste linear	Ajuste quadrático
Todos os Blocos	2,01	1,75
Seleção ₁	2,35	2,17
Seleção ₂	2,52	2,37

Tabela 2 - erros quadráticos médios dos melhores ajustes sobre os dados de segmentação manual, para três tipos de seleção de blocos.

Apesar dos erros médio quadráticos encontrados serem baixo para um bloco, eles podem ser vistos como altos quando o problema em questão é estimar o número de ovos de uma palheta, cujo número de blocos de imagens varia entre 80 e 140.

Além disso, percebeu-se que o limiar de binarização é um fator muito importante para a qualidade dos ajustes. Para as saídas reais no intervalo $[0,1]$, um limiar de binarização de 0,5 significa que uma resposta da rede menor que 0,5 é rotulada como OVO enquanto uma resposta acima de 0,5 é rotulada como FUNDO. Assim, cada rede deve ter um limiar ótimo que minimize o erro quadrático médio dos ajustes.

6.2.2. RBF

Afim de obter os melhores resultados, procurou-se para cada conjunto de dados, e para cada seleção de blocos, os limiares cujos erros quadrático médio sobre os ajustes fossem o mínimo.

A Tabela 3 dá as estimativas de contagem do conjunto C1 para o conjunto das nove palhetas e com os blocos da seleção₂. O melhor limiar de binarização encontrado foi de 0,0078, que resultou num erro quadrático médio de ajuste de 4,16 ovos, para um ajuste linear. O erro relativo médio e seu desvio padrão, calculados com os dados da Tabela 3, foram respectivamente 35% e 13%.

Seleção de blocos: Seleção₂		Ajuste: linear	Limiar: 0.0078	Erro quadrático médio de ajuste: 4,16 ovos
Palheta	Nº de ovos	Nº de ovos estimado		Erro relativo(%)
F1-026	197	153		0,22
F1-040	146	55		0,62
F1-034	293	212		0,27
F2-094	386	216		0,44
F2-070	373	224		0,39
F2-046	429	291		0,32
F3-045	594	392		0,34
F3-047	1140	980		0,14
F3-083	1182	733		0,37

Tabela 3 - Estimativas do número de ovos nas palhetas simuladas com uma rede RBF para palhetas descritas pelo conjunto de C1, com a seleção de blocos Seleção₂.

A mesma análise feita agora para o Conjunto C2, com a mesma seleção de blocos Seleção₂, tem seus resultados das estimativas do número de ovos nas palhetas na Tabela 4. O limiar de binarização encontrado foi o mesmo, 0.0078, com um erro quadrático médio de 12,80 ovos por palheta para um ajuste linear. O erro relativo médio e seu desvio padrão, calculados com os dados da Tabela 4, foram respectivamente 83,8% e 2%.

Seleção de blocos: Seleção₂		Ajuste: linear	Limiar: 0.0078	Erro quadrático médio de ajuste: 12,80 ovos
Palheta	Nº de ovos	Nº de ovos estimado		Erro relativo(%)
F1-026	185	36		0,80
F1-040	290	47		0,83
F1-034	160	30		0,81
F2-094	1433	175		0,87
F2-070	404	60		0,85
F2-046	133	27		0,79
F3-045	646	90		0,86
F3-047	370	56		0,84
F3-083	1312	189		0,85

Tabela 4 - Estimativas do número de ovos nas palhetas simuladas com uma rede RBF, com o conjunto de dados C2, com a seleção de blocos Seleção₂.

A Tabela 5 mostra as estimativas de contagem de todas as nove palhetas para Conjunto C2_{repr} com blocos selecionados pela Seleção₂. O limiar de binarização encontrado foi de 0.023, que resultou num erro quadrático médio de ajuste de 13,9 ovos para um ajuste quadrático. O erro relativo médio e seu desvio padrão, calculados com os dados da Tabela 5, foram respectivamente 92,1% e 3,2%.

Seleção de blocos: Seleção₂		Ajuste: quadrático	Limiar: 0.023	Erro quadrático médio de ajuste: 13,9 ovos
Palheta	Nº de ovos	Nº de ovos estimado		Erro relativo(%)
F1-026	197	18		0,90
F1-040	146	0		1
F1-034	293	26		0,91
F2-094	386	29		0,92
F2-070	373	28		0,92
F2-046	429	40		0.90
F3-045	594	50		0,91
F3-047	1140	135		0,88
F3-083	1182	98		0,91

Tabela 5 - Estimativas do número de ovos nas palhetas simuladas com uma rede RBF para palhetas descritas pelo conjunto de dados C2_{repr}.

Uma vez que a retirada de blocos é um processo custoso e sujeito a critérios subjetivos, faz-se necessário avaliar se sua realização surtirá benefícios à melhora na estimativa da contagem. A Tabela 6 trás os resultados das estimativas de contagem para o conjunto de dados C1, com todos os blocos. O limiar encontrado foi de 0,023 e o erro quadrático médio de ajuste foi de 12,12 ovos, para um ajuste quadrático. O erro relativo médio e seu desvio padrão, calculados com os dados da Tabela 6, foram respectivamente 66% e 19%. Um teste de hipótese de paridade com significância de 5% foi realizado entre os erros relativos dos conjuntos C1-C1_{repr} representados nas Tabela 3 e Tabela 6, onde um deles possui seleção de blocos Seleção₂, e o outro analisa todos os blocos. A hipótese nula foi rejeitada, mostrando que os erros quadráticos médios são diferentes estatisticamente. Ou seja, o erro quadrático médio da estimativa dos ovos nas palhetas, é menor para o conjunto C1_{repr} em comparação ao Conjunto C1.

Seleção de blocos: Todos		Ajuste: quadrático	Limiar: 0.273	Erro quadrático médio de ajuste: 12,12 ovos
Palheta	Nº de ovos	Nº de ovos estimado		Erro relativo(%)
F1-026	230	85		0,63
F1-040	255	104		0,59
F1-034	328	145		0,55
F2-094	442	55		0,87
F2-070	449	86		0,80
F2-046	543	81		0,92
F3-045	761	142		0,93
F3-047	1235	897		0,95
F3-083	1299	485		0,96

Tabela 6 - mesmo que a Tabela 53, porém todos os blocos foram selecionados.

A análise estatística para redes RBF pode ser sintetizada da seguinte forma:

- A estimativa da quantidade de ovos com o conjunto de dados $C1_{repr}$, gera os melhores resultados, visto que nos conjuntos $C2$ e $C2_{repr}$ a rede capturou ainda mais ruídos, levando a uma média de erros muito alta, gerando resultados sem relevância para uma contagem confiável
- Na situação onde ocorre a exclusão de blocos ruidosos (Seleção₂), pode-se afirmar que o ajuste sobre o conjunto de dados $C1_{repr}$ gera estimativas mais precisas que aquelas obtidas com o ajuste sobre o conjunto de dados $C1$, onde estão presente todos os blocos.
- Para os processamentos com os conjuntos $C2$ e $C2_{repr}$ existe tanto ruídos que não se deve considerar seus resultados.
- Mesmo com uma taxa média de erro de 35% o conjunto $C1_{repr}$ foi o que apresentou melhores resultados.

6.3 Problemas e soluções

Analisando os resultados obtidos e com base nas imagens originais e com as segmentações manuais feitas anteriormente detectamos alguns problemas nos testes realizados.

A rede estava muito sensível aos ruídos encontrados nas imagens das ovitrampas. Esta sensibilidade fazia com que a rede fizesse uma contagem errada da quantidade de ovos em cada ovitrampa, pois devido a tonalidade dos ruídos ser semelhante a dos ovos, a rede interpreta os ruídos como ovos. Duas soluções foram testadas, primeiro a adoção do viés no treinamento, que já foi explicado anteriormente como se dá seu funcionamento. O uso do viés melhorou a taxa de acerto da rede com relação aos padrões de fundo, que foi verificado e explicado ser mais crítico para o desempenho da rede.

Num outro instante foi testada uma solução que não gerou os resultados esperados. Esse teste será descrito abaixo em tópicos:

1. Após o treinamento da rede, feito com padrões extraídos de imagens com padrão de cor RGB, as imagens das ovitrampas foram passadas pela rede e suas respostas guardadas.
2. A rede foi novamente treinada com um novo conjunto de treinamento, agora os dados de treinamento foram extraídos de imagens com 256 tons de cinza.
3. As respostas geradas pela rede foram transformadas em imagens em 256 tons de cinza e passadas novamente pela rede.
4. A resposta da rede foi verificada, e foram observadas respostas piores para os valores de contagem dos ovos. E após a geração das imagens, verificou-se que em alguns pontos as imagens ficaram mais limpas, porém na maior parte da imagem os ruídos ficaram mais intensos.

Outro problema é a questão do tempo de processamento das imagens. A rede, devido as implementações de I/O, estão demorando um tempo ainda longo para testar a imagem e gerar a saída.

7. Conclusões e Trabalhos Futuros

O sistema desenvolvido visa facilitar o trabalho realizado pela secretaria de saúde na realização da contagem de ovos das armadilhas expostas por diversas áreas da cidade. Com um resultado determinado pela rede sendo próximo ao valor real da quantidade de ovos, o sistema será confiável para a realização do trabalho, colaborando com o projeto de contagem dos ovos.

O trabalho realizado permitirá a expansão do projeto Sistema de Monitoramento e Controle Populacional do Vetor da Dengue(SMCP-Aedes), que hoje ainda abrange poucas regiões. Agora com a tecnologia do sistema automático é preciso uma infra-estrutura mais acessível tanto de equipamentos quanto de pessoal, o que facilita a expansão do projeto para uma quantidade maior de cidades.

O tecnologia utilizada mostrou-se satisfatória no desenvolvimento do projeto à medida que foi correspondendo as expectativas geradas no início, na idéia da concepção do projeto.

Os resultados obtidos foram satisfatórios na primeira etapa do projeto, porém alguns trabalhos podem ser realizados afim de melhorar os resultados, tais como:

- A avaliação será feita diretamente das imagens, sem ser preciso extrair as características e gerar um documento de texto para ser a entrada das redes.
- Fazer um pré-processamento das imagens afim de diminuir os ruídos das imagens que estão gerando problemas na contagem dos ovos.
- Análise de novos modos de definição de raios, pois foi verificado que eles tem grande influência na qualidade dos resultados
- Realizar no algoritmo mudanças que permitam diminuir o tempo de execução, ou seja, o tempo para realizar a contagem.

- Fazer uma análise afim de obter informações sobre porque os conjuntos $C2$ e $C2_{repr}$ geraram resultados tão ruins, onde o esperado era uma melhora nas estimativas.
- Introduzir no algoritmo uma análise de forma dos ovos, pois dessa forma poder-se-ia evitar o erro que o algoritmo está cometendo quando confunde os ruídos e sujeiras com ovos.

8. Referências Bibliográficas

[1] **Redes Neurais - A Rede RBF** – Online, acesso em 20/04/2010 na url <http://www.inf.ufrgs.br/~engel/Common/CMP121/RBF.pdf>

[2] **Redes de Funções de base Radial** – Online, acesso em 20/04/2010 na url http://www.cin.ufpe.br/~gcv/web_lci/Aula-RBFs-sem-fundo.pdf

[3] **Homepage de Redes Neurais** – Online, acesso em 20/04/2010 na url <http://www.din.uem.br/ia/neurais/>

[4] **Aprendizado** – Online, acesso em 20/04/2010 na url http://www.lncc.br/~labinfo/tutorialRN/frm1_aprendizado.htm

[5] **Rede Neural** – Online, acesso em 20/04/2010 na url http://pt.wikipedia.org/wiki/Rede_neural

[6] **What's k-Mean Clustering?** – Online, acesso em 20/04/2010 na url <http://people.revoledu.com/kardi/tutorial/kMean/WhatIs.htm>

[7] **The Backpropagation algorithm** - Online, acesso em 10/01/2010 na url <http://www.speech.sri.com/people/anand/771/html/node37.html>

[8] **K-means clustering** - Online, acesso em 20/04/2010 na url http://en.wikipedia.org/wiki/K-means_clustering

[9] **Análise de agrupamentos** – Online, acesso em 20/04/2010 na url <http://www.inf.ufsc.br/~patrec/agrupamentos.html>

[10] **Reconhecimento de Padrões** – Online, acesso em 20/04/2010 na url <http://ltdi.est.ips.pt/jascenso/padroes/teoricas/Aula%207%20-%20Aprendizagem%20Não%20Supervisionada.pdf>

[11] **Redes Neurais** – Online, acesso em 20/04/2010 na url http://www.cin.ufpe.br/~gcv/web_lci/Aula-introducao-redes-neurais.PDF

[12] **Rede de Funções de Base Radial (RBF)**- Online, acesso em 20/04/2010 na url <http://www.deti.ufc.br/~guilherme/TIP705/rbf.pdf>

[13] **Redes Neurais Artificiais: Conjuntos Fuzzy e redes RBF** – Online, acesso em 20/04/2010 na url http://www.deti.ufc.br/~guilherme/PAPERS/COELCE-2_mod-2.pdf

[14] **Radial Basis Function** – Online, acesso em 20/04/2010 na url http://www.ee.pucrs.br/~decastro/RNA_hp/pdf/RNA_C5.pdf

[15] **Redes neurais Artificiais** – Online, acesso em 20/04/2010 na url <http://www.icmc.usp.br/~andre/research/neural/index.htm>

[16] **Introdução às Redes Neurais Artificiais** – Online, acesso em 20/04/2010 na url <http://www.inf.ufsc.br/~barreto/tutoriais/Survey.pdf>

[17] **Implementação do algoritmo K-means** – Online, acesso em 20/04/2010 na url <http://200.134.81.19:8080/gpes/arquivos/material/inteligencia/desenvolvimento/relatorios/2009/D.POZZO.2009.A.pdf>

[18] **Modelo do Neurônio Artificial** – Online, acesso em 20/04/2010 na url http://www.lncc.br/~labinfo/tutorialRN/frm1_neuronio.htm

[19] **Redes Neurais Artificiais** – Online, acesso em 30/03/2010 na url http://www.ppgia.pucpr.br/~euclidesfjr/Metodos_Inteligentes/0206/RNA-XIIERI.pdf

[20] Ludermir, T. B. , Carvalho, A.C.P. , Braga, A.P., **Redes Neurais Artificiais – Teoria e aplicações**. Livros Técnicos e Científicos Editora, 2000

[21] Simon Haykin, **Redes Neurais – Princípios e prática**. Artmet Editora S.A. 2002

[22] **Introduction of the Radial Basis Function(RBF) Networks** – Online, acesso em 20/04/2010 na url <http://www-users.cs.york.ac.uk/~adrian/Papers/Others/OSEE01.pdf>

[23] **Redes Neurais – Redes de Função de base Radial** – Online, acesso em 20/04/2010 na url http://xa.yimg.com/kg/groups/18952383/35890612/name/RBF_1.pdf

[24] **Dicas para configuração de redes neurais** - Online, acesso em 20/04/2010 na url <http://rfaces.googlecode.com/files/0D81DB6Ed01.pdf>

[25] **Aplicações de redes neurais artificiais em simulações** – Online, acesso em 20/04/2010 na url www.simpep.feb.unesp.br/.../copiar.php?...Aplica%E7%F5es%20de%20redes%20neurais%20artificiais%20e

[26] **Multilayer Perceptron** – Online, acesso em 20/04/2010 na url http://www.ee.pucrs.br/~decastro/RNA_hp/pdf/RNA_C4.pdf

[27] Regis, L., Monteiro, A.M., Melo-Santos, M.A.V., Silveira Junior, J.C., Furtado, A.F., Acioli, R.V., Santos, G.M., Nakazawa, M.M., Carvalho, M.S., Justiniano Jr, P., Souza, W. Developing new approaches for detecting and preventing Aedes aegypti population outbreaks: basis for surveillance, alert and control system. Mem. Instituto Oswaldo Cruz (submetido).

[28] McClelland, J. , Rumelhart, David E. and The PDP Research Group, ***Parallel Distributed Processing - Exporations in the Microstructrures of Cognitions***, vol2, 1999.

[29] ZSolt L Kovacs, **Redes Neurais Artificiais: Fundamentos e Aplicações**, Editora Livraria da Física, 2006