
Grupo 3

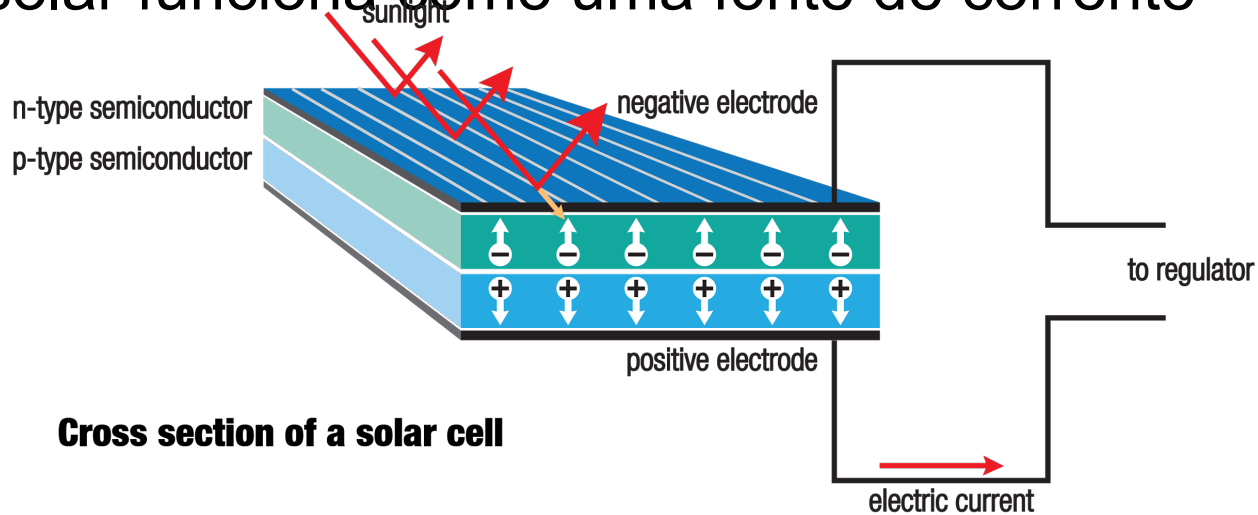
- Vincent Gliniewicz
 - Bruno Giordano
 - Estevão Vieira
-

-
- 1. Estimativa de produção de energia solar**
 - 2. Reconhecimento de objetos**
 - 3. Reconhecimento de Folhas**
-

Estimativa de produção de energia solar

Um pouco de teoria

Painéis fotovoltaicos trabalham como PN-diodos e a luz solar funciona como uma fonte de corrente



Teoria & parametros

- A corrente produzida depende da intensidade da luz, mas também da temperatura do painel fotovoltaico: a produção diminui sobre um dada temperatura.
- A intensidade da luz é dada por dois parametros:
 - luz direta (mais forte, mas direcional)
 - luz dissipada (menos forte mas mais espalhadas)
- A temperatura do painel depende de três parametros:
 - A temperatura ambiente (resfria o painel)
 - O vento (aumenta convecção, ajuda resfriamento)
 - A energia recebida (aquece o painel)

Suposições para facilitar o modelo

- O azimute depende apenas da hora do dia (não de as estações do ano)
- O efeito do ângulo vertical é desprezível
- O painel é sempre limpo
- Toda a energia produzida é consumida (utilização de bateria).
- A carga não influencia a produção
- Todos os outros parâmetros são desprezíveis

Dados

- Os dados são recolhidos em Estocolmo e Phoenix nos painéis solares 20kW (dadas por meu professor na Suécia). Todas horas foram medidas:
 - Temperatura ambiente (C)
 - Beam Irradiance (W/m^2)
 - Diffuse Irradiance (W/m^2)
 - Velocidade do vento
 - produção média de energia (Wh/h)
- A posição do sol foi derivada a partir da hora do dia

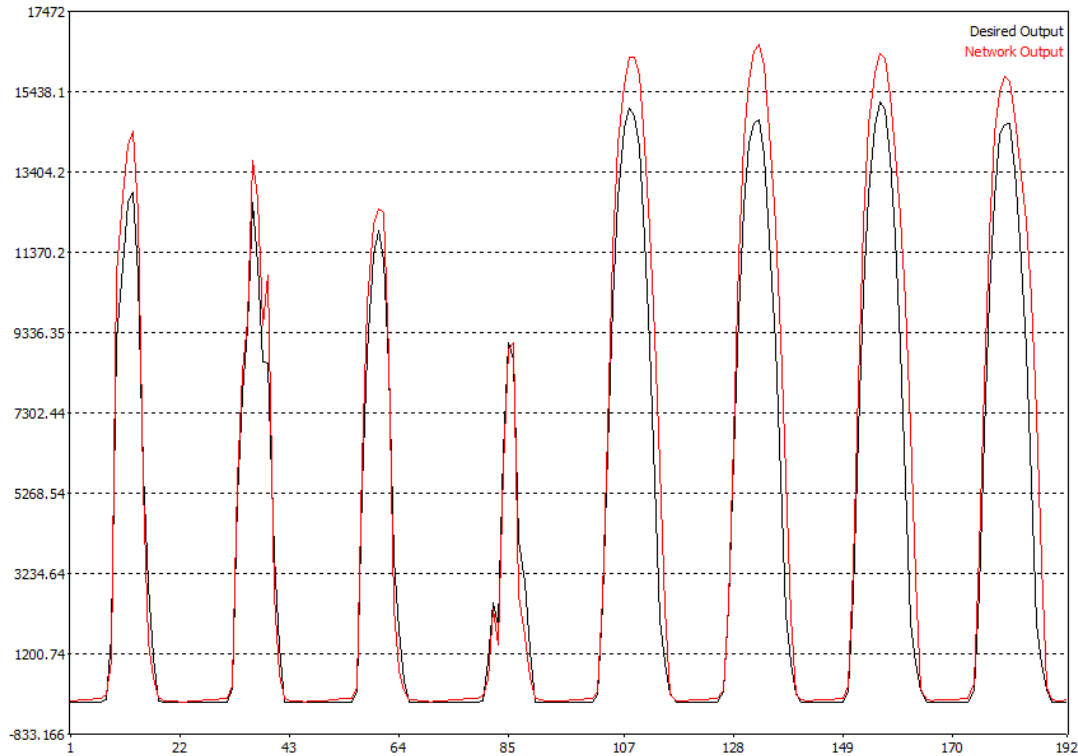
Dados - 2

- Estocolmo : Painel posicionado para o sul (100% - 13horas)
- Phoenix : Painel posicionado para o leste (100% - 8 horas)
- Treinamento : 30 dias em janeiro para cada cidade
- Teste: 30 dias em junho para cada cidade

Dados - exemplo

Azimute	Beam Irradiance	Diffuse irradiance	Temperatura	Vento	Energia produiza
0.722222222	0	0	6	9.6	0
0.791666667	0	17	5.2	8.7	198.932
0.861111111	30	45	5.2	8.4	1176.656
0.930555556	225	44	5.2	8	2977.659
1	159	49	5.2	7.7	2441.095
0.930555556	114	31	4.6	8.2	1548.116
0.861111111	0	10	4.1	8.8	64.983

Resultados - redes 5-1 linear

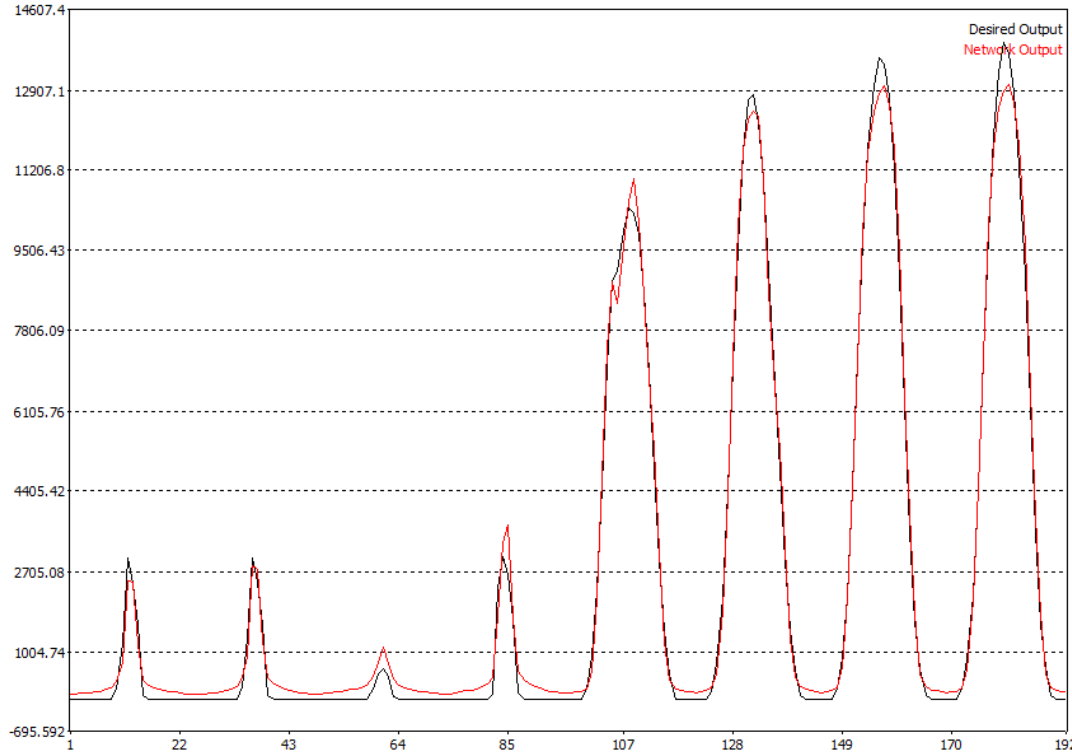


RMS erro:

- 0.057 para treinamento
- 0.11 para teste

A simulação não converge

Resultados - redes 5-2-1 sigmoid



RMS erro:

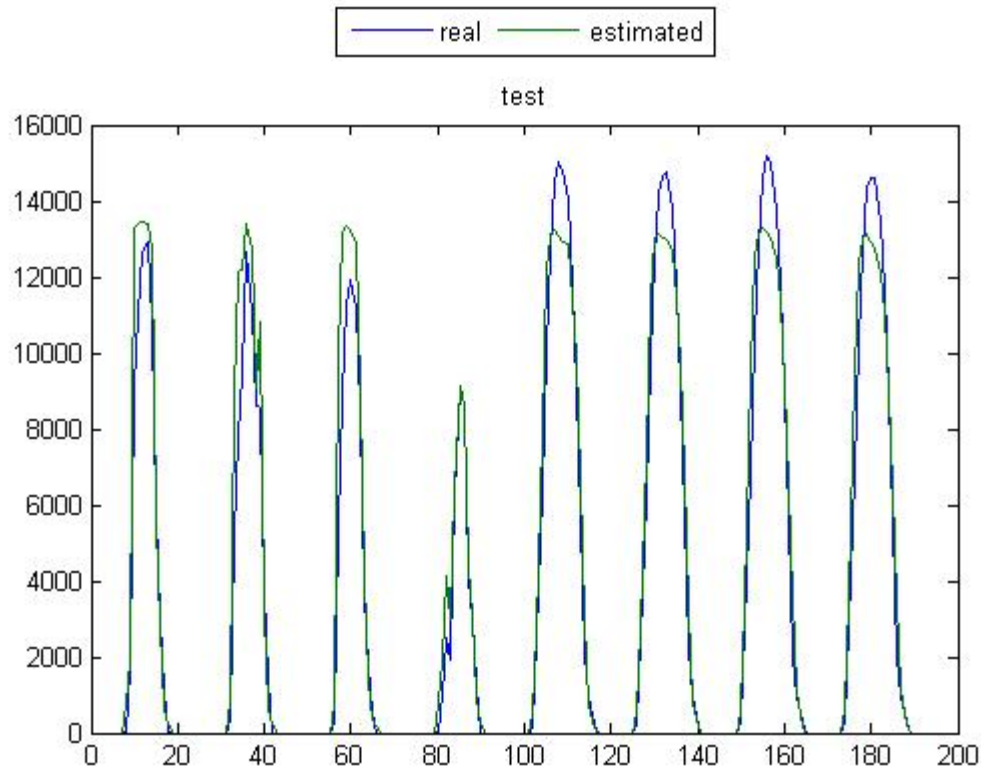
- 0.01 para o treinamento
- 0.04 para o teste

A simulação converge em 100 iterações.

Problemas

- 0 produção significa apenas que nenhuma luz solar está atingindo o painel (os outros parâmetros são irrelevantes neste caso).
- Este problema é resolvido ao fazer um pré-processamento: 0W/m^2 de intensidade solar significa 0W de produção.

Rede 5-2-1 sigmoid + processamento



RMS erro:

- treinamento: 10% melhoria
- teste: 25% melhoria

Refinamento

Randomização de dados em conjuntos de treinamento e teste:

- loop para obter o melhor mínimo local
- Aumenta a complexidade.
- Loop para fazer k-fold cross validation

Refinamento

- Usando MATLAB, observamos 10 resultados diferentes ao longo de 10 loops. O melhor resultado é mantido (o melhor minimum local).
- Overfitting comportamento é atingido com 41 neurônios. A melhor rede deve ser menor.

Cross Validation

- A rede mais flexível é de alguma forma melhor do que aquela que dá o melhor resultado apenas uma vez.
- Observado: por 10 dadas conjuntos de treinamento, 24 neurônios dar o resultado mais estável. (e muito bom: $R = 0,9993$)

Embarque



- Um dispositivo de medição inteligente pode fazer a previsão
- A rede primeiramente treinará com dados padrão e será treinada novamente (uma semana depois) com dados próprios.
- kWh medido diretamente com dispositivo
- dados meteorológicos obtidos através do site: solarGIS.info mas é caro...

Conclusões

- Rede bastante eficiente com poucos de entrada (5) e processamento.
- Deve ser fácil de embarcar, embora os dados de previsão são caros.
- Não é bom para uso privado, mas parece ser uma boa solução para uma empresa.

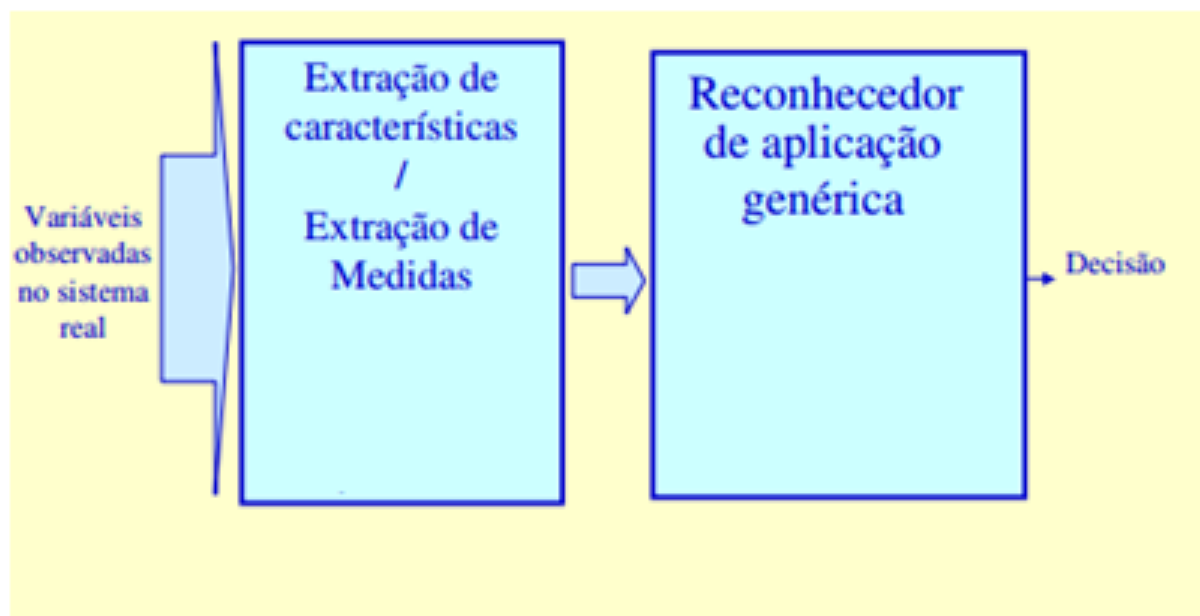
Reconhecimento (categorização) de Objetos

Aplicações:

- Internet of Things (IoT)
- Robótica doméstica
- Busca de imagens na internet (google)
- Sensores inteligentes

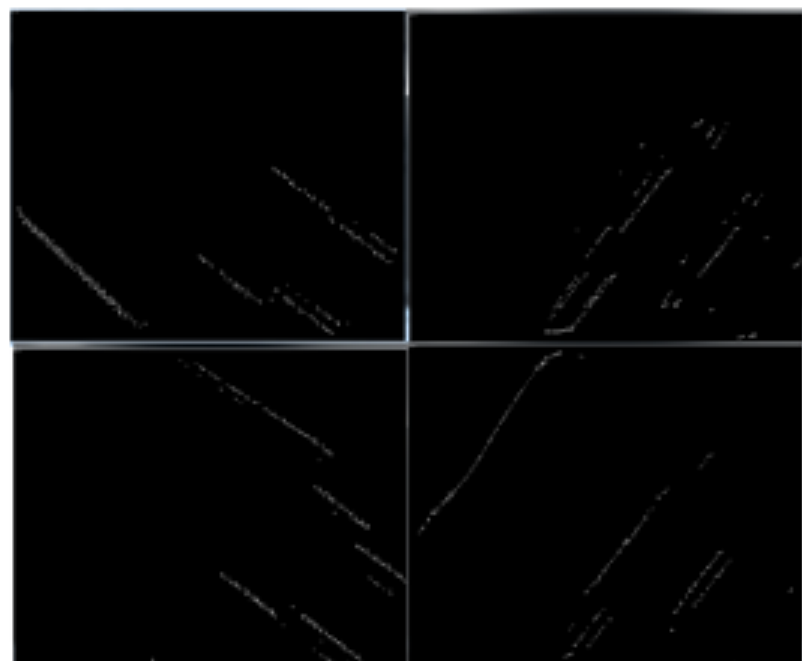
Estratégia utilizada:

- Solução em dois blocos, extrator de características + rede neural:



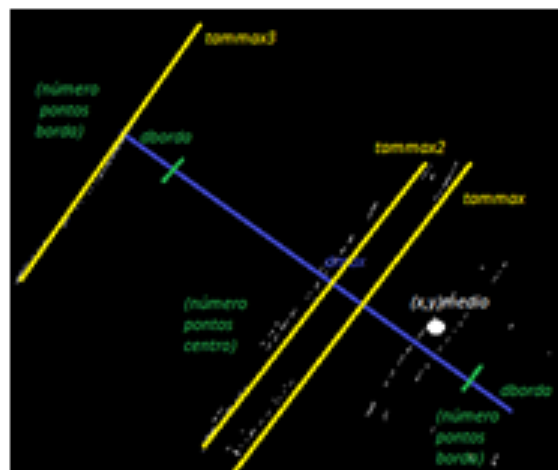
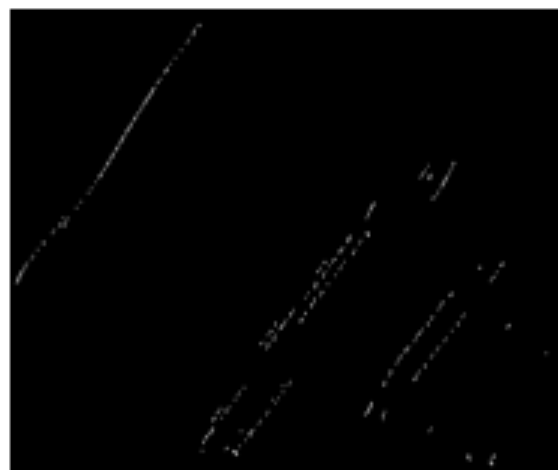
Extração de características

- Baseando-se no algoritmo de Canny para detecção de bordas da imagem, organizando os pontos definidos como borda de acordo com os ângulos na matriz de gradiente da imagem, é possível decompor uma imagem em grupos de retas paralelas de maneira extremamente simples:



Extração de características

- A partir de cada grupo de reta paralela, definidos pelo ângulo médio do gradiente dos pontos do grupo, foram organizados de acordo com a quantidade de pontos em cada grupo (grupo 1 é o grupo com mais pontos, grupo 2 é o segundo grupo com mais pontos, etc.):



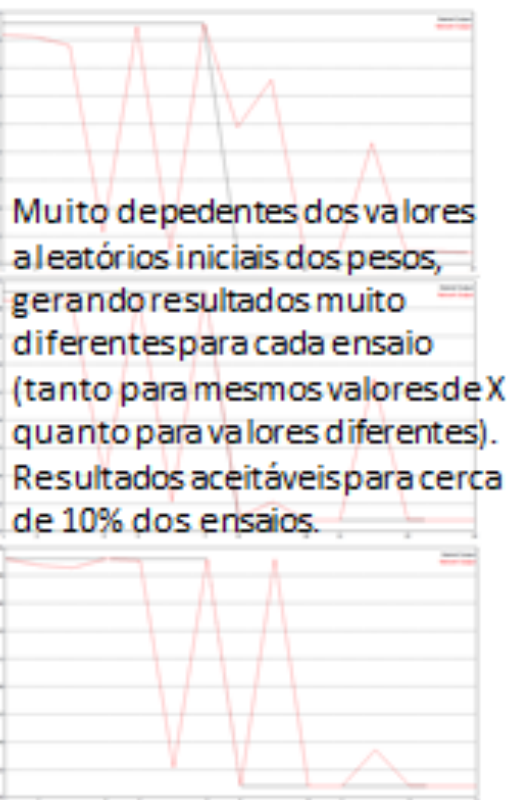
11 Parâmetros:

- Distância entre os (x,y) médio dos dois grupos com maior número de pontos, normalizado pela média dos d_{max} de cada grupo
- Diferença de gradiente médio entre o 1 e o 2 grupo, como também entre o 2 e o 3 grupo
- Proporção entre o número de pontos considerados como borda/central para cada grupo
- Proporção entre os três maiores tamanhos de retas, exemplificado na figura como $tammax2/tammax$ e $tammax3/tammax$, para cada grupo

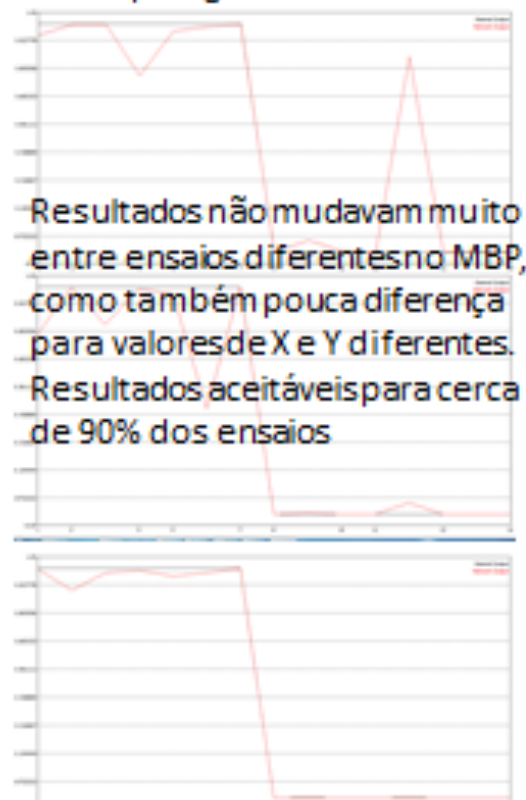
Reconhecedor (rede neural)

- Para a definição da topologia da rede neural, foi utilizado o software Multiple Back-Propagation, realizando diversos ensaios e verificando a porcentagem de resultados aceitáveis para cada uma das topologias abaixo:

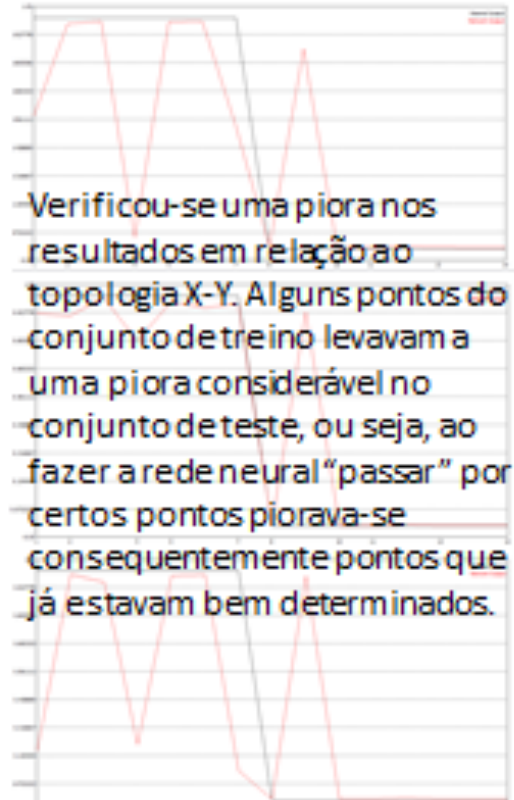
Topologias X-1



Topologias X-Y-1

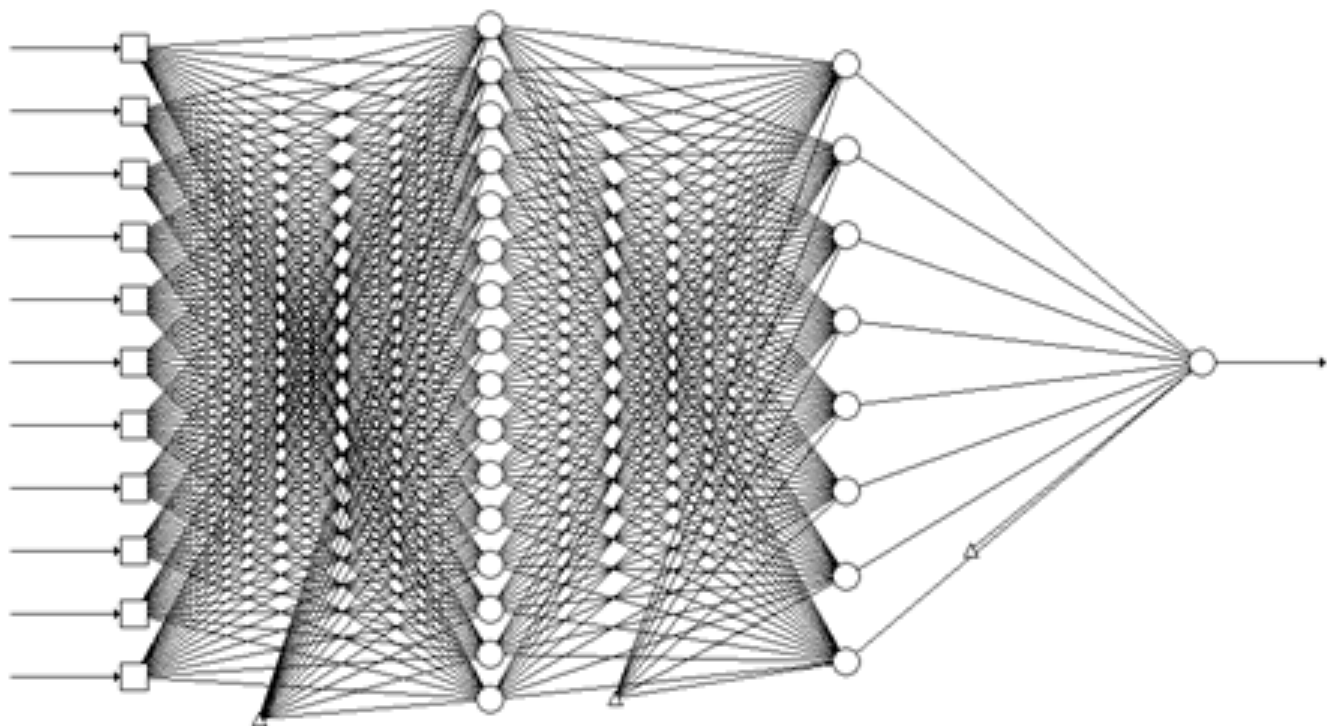


Topologias X-Y-Z-1



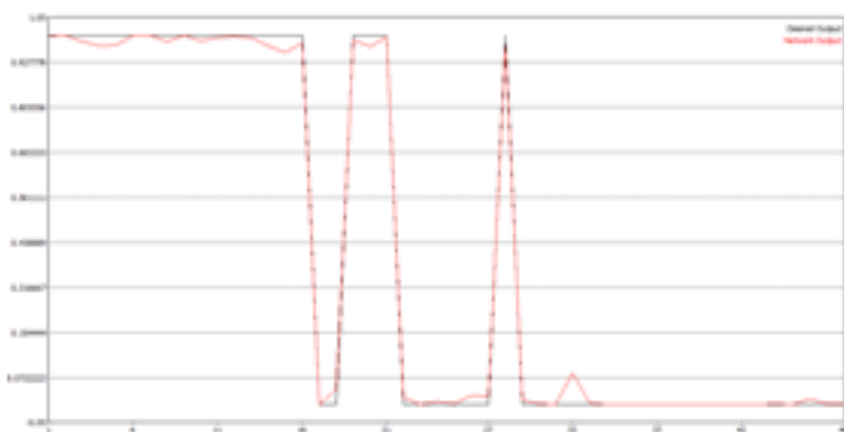
Reconhecedor (rede neural)

Utilizando a topologia X-Y-1, modificamos os valores de X e Y, realizando vários ensaios para cada valor (X,Y) determinado, comparando os resultados. Embora todos os resultados fossem extremamente parecidos mesmo para valores de X e Y diferentes, foi verificado uma otimização para a topologia 16-8-1 (convergências rápidas e resultados sempre aceitáveis):

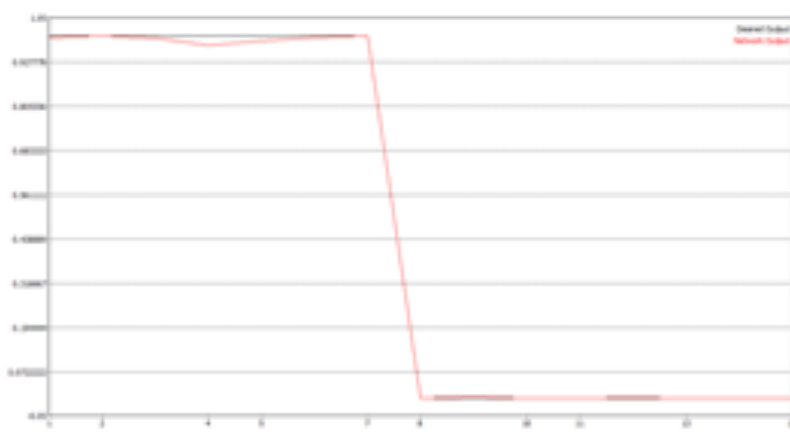


Reconhecedor (rede neural)

Treino



Teste



RMStreino: 0.0095

RMSteste: 0.0038

Taxa de Acerto: 100% (70 imagens)

Inicialmente, os conjuntos de treino/teste foram definidos aleatoriamente. Fazendo ensaios no MBP, os pontos do conjunto de teste fora da curva eram trocados com pontos do conjunto de treino, para que a rede neural pudesse corrigir esses pontos, resultando nos conjuntos indicados acima.

Programa (em C++)

```
Digite o caninho+nome da imagen (exemplo: C:\Users\imagen.bmp):C:\\\imagens\\3.bmp
p
Tempo de execucao do algoritmo, em ms (sem incluir o carregamento da imagen): 62

INPUTS: 0.124654;0.611279;0.388721;0.499714;0.927214;0.39875;0.378423;0.629577;0.
.41895;0.876358;0.336687;

Eh un carregador!
(output: 0.999993)
Press <RETURN> to close this window...
_
```



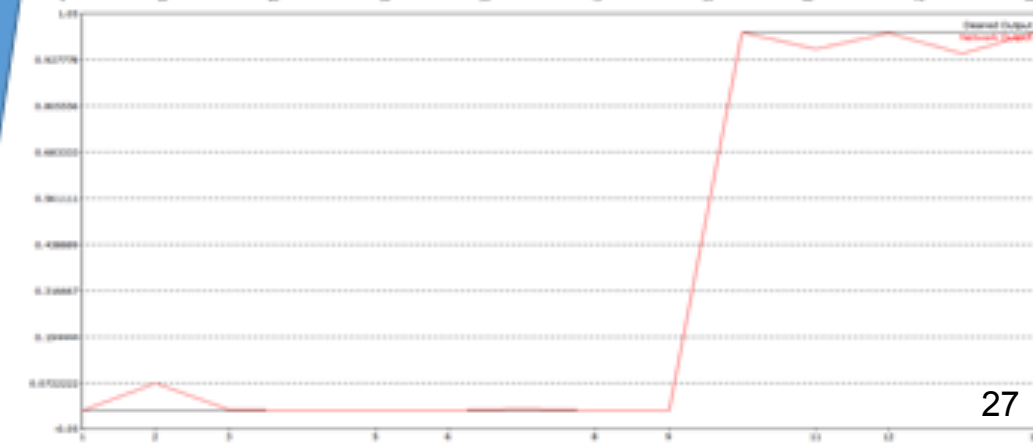
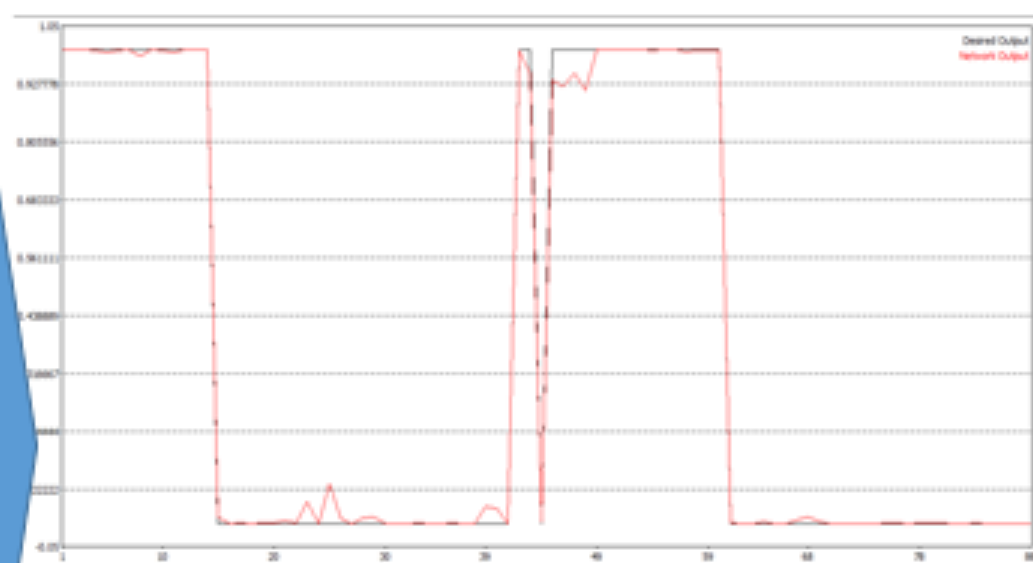
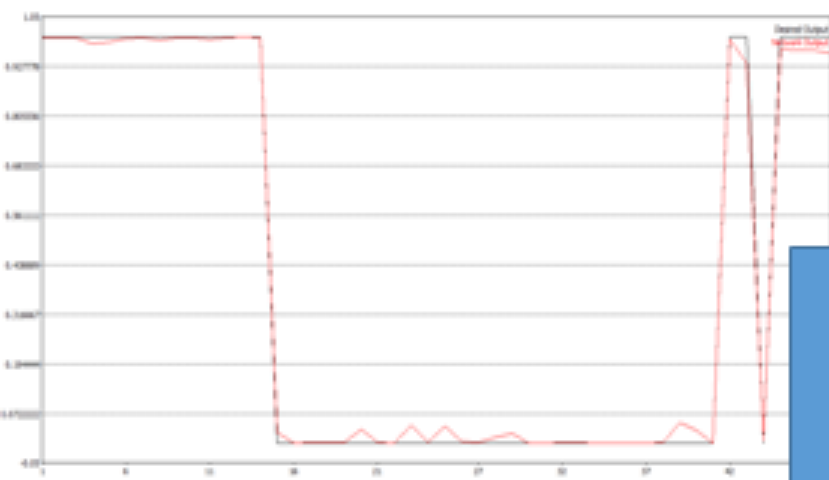
```
Digite o caninho+nome da imagen (exemplo: C:\Users\imagen.bmp):C:\\\imagens\\4.bmp
p
Tempo de execucao do algoritmo, em ms (sem incluir o carregamento da imagen): 93

INPUTS: 0.8779269;0.177743;0.822257;0.466279;0.951875;0.898883;0.517223;0.482777;
0.231595;0.887875;0.713577;

Nao eh un carregador!
(output: 0.71146e-005)
Press <RETURN> to close this window...
_
```



Inclusão de novo objeto na rede



Análise das qualidades

- A categorização de padrões em imagens usualmente utiliza todos os pixels da imagem, com um número enorme de variáveis de entrada e neurônios (consequentemente com conjuntos de treino e teste enormes também), impactando no tempo de execução impossibilitando aplicações em tempo real para hardware limitado. A abordagem proposta forneceu resultados satisfatórios para apenas 11 variáveis de entrada e 25 neurônios ($16+8+1$), com um tempo de execução viável para aplicações em tempo real com hardware limitado (robótica)
- As imagens utilizadas foram em grande parte retiradas da internet, ou seja, objetos da mesma classe (no caso, carregadores) mas com ligeiras diferenças geométricas e mesmo assim tivemos resultados satisfatórios.

Análise dos defeitos

- A implementação do algoritmo de Canny para definição das bordas foi feita inteiramente em C++ sem auxílio de bibliotecas consolidadas. Essa abordagem possibilitou o entendimento do algoritmo e consequentemente a utilização de resultados intermediários importantes, mas também fez com que essa funcionalidade não esteja perfeitamente implementada impactando nos resultados
- Objetos metálicos e transparentes são extremamente problemáticos

Possíveis Otimizações

- Só foram considerados os dois grupos de retas paralelas com maior número de pontos. Uma otimização imediata seria aumentar para 3, 4, 5, (...), n , grupos de retas paralelas, possibilitando uma melhor descrição de objetos mais complexos
- A extração dos parâmetros de cada grupo de retas paralelas pode ser otimizada levando em consideração mais parâmetros além dos descritos anteriormente
- Resultados semelhantes em termos de acertos para conjunto de treino e teste não implicam em redes neurais semelhantes. Nos diversos ensaios realizados no MBP foi percebido a diferença nos pesos sinápticos da rede neural. Consequentemente, é possível implementar N redes neurais diferentes e tirar conclusões baseado nas N decisões de cada uma delas (por exemplo, se todas as N redes indicarem o mesmo resultado aceitar o resultado, caso contrário tratar o dado incluindo-o no conjunto de treino das N redes)
- Não foram levadas em consideração as cores do objeto, que podem gerar resultados ainda melhores se incluídas

Conclusão

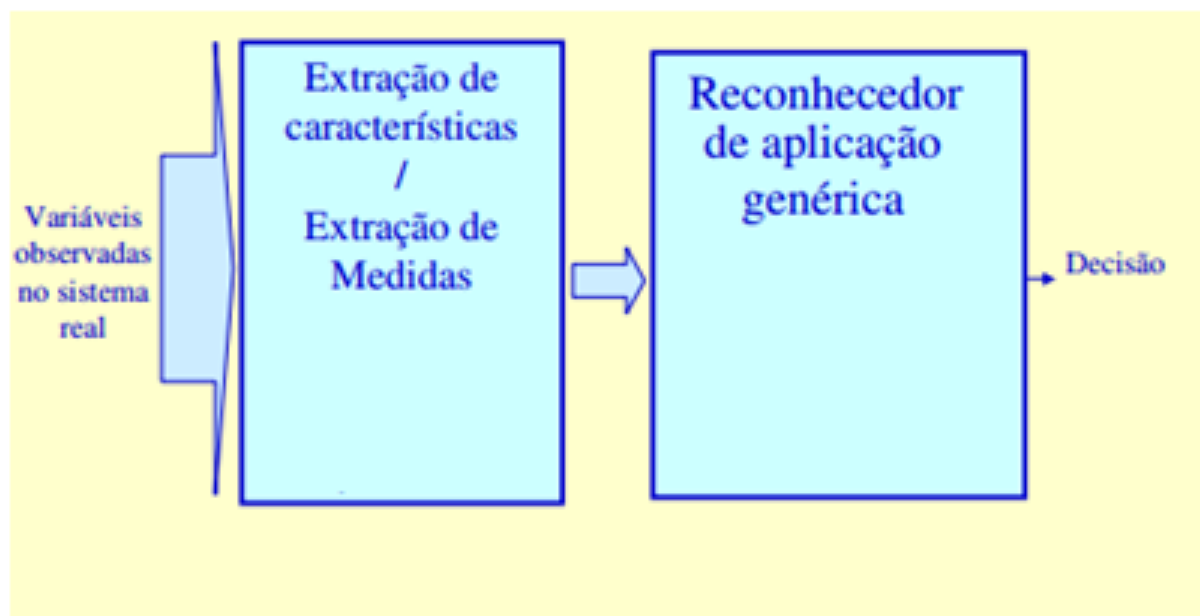
A disciplina foi extremamente esclarecedora na introdução da utilização de redes neurais para solução de problemas de reconhecimento de padrões, em particular à imagens, possibilitando uma base interessante para possíveis aprofundamentos visando a solução de problemas desafiadores.

Reconhecimento de Folhas

Aplicações desde pesquisa científica à coleta de ervas para indústria.

Estratégia utilizada:

- Solução em dois blocos, extrator de características + rede neural:



Coleta das folhas

- Um total de 210 folhas foram coletadas, de pelo menos 20 espécies diferentes.
- No trabalho final, apenas 104 foram utilizadas, provenientes de 12 espécies.
- A diminuição se deu em respeito às restrições adicionais que surgiram sobre as imagens.

As folhas

I



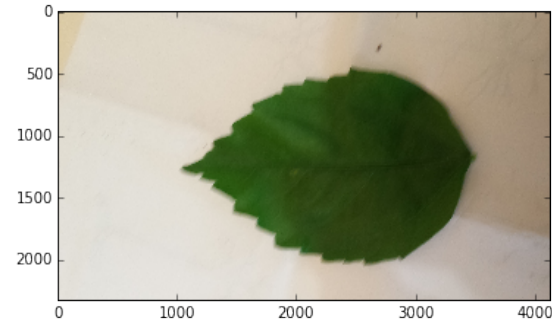
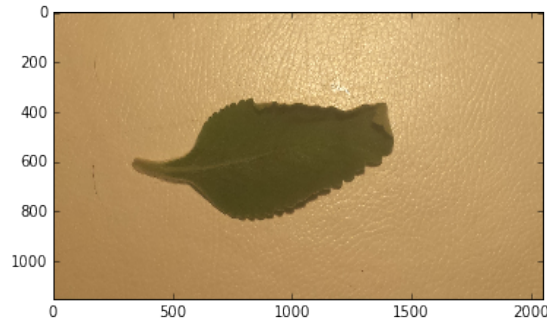
As Medidas

- 7 Parâmetros Normalizados
- Todos derivados das medidas diretas
 - Área
 - Perímetro
 - Diâmetro
 - Altura Fisiológica
 - Largura fisiológica

O pré-processamento

I

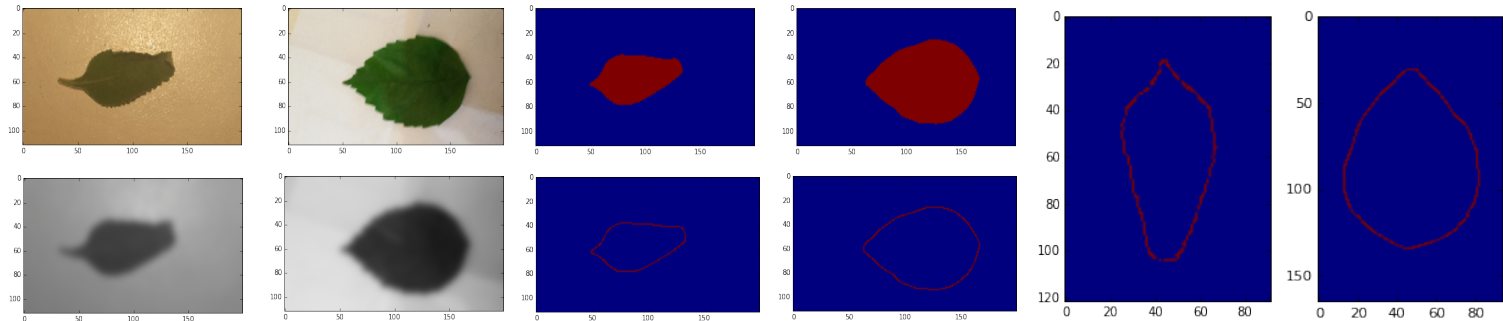
- Tentativa de automatização completa
- Necessidade de criação de restrições
 - Folha centralizada
 - Fundo branco



O pré-processamento

II

- Custo computacional elevado

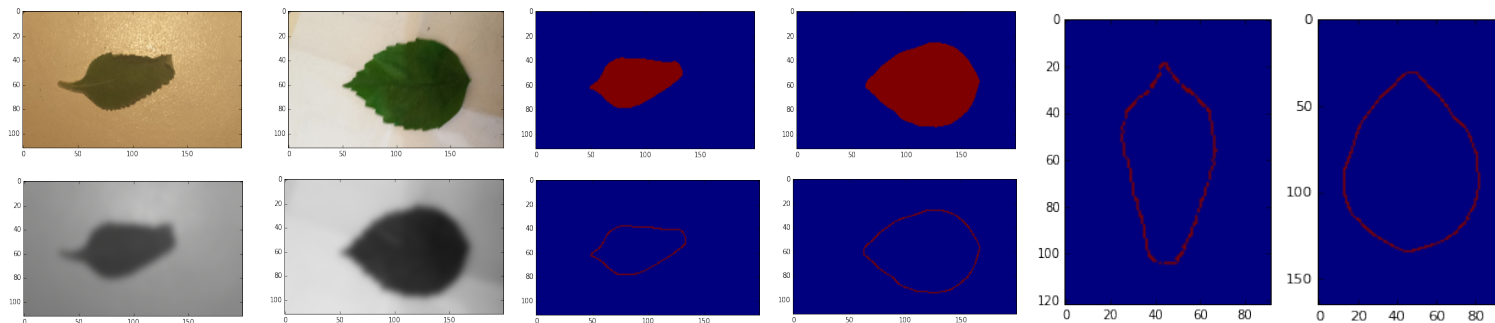


O pré-processamento

III

- Custo computacional elevado
- “Bonito, porém ineficiente”

-Gliniewicz, V.



As folhas

II

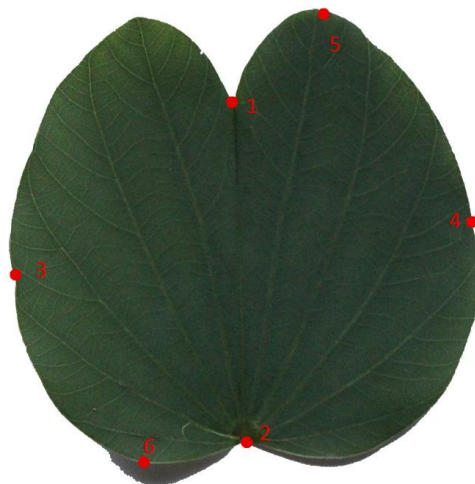


- Interação necessária com o usuário.
- Automatização era muito complexa para trabalho muito simples.
- Usuário não precisa de nenhum conhecimento para a interação.
- 6 Cliques.

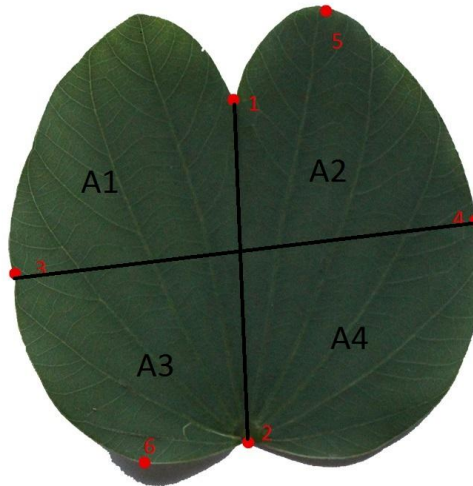
As Medidas



- 13 Parâmetros Normalizados
- Todos derivam de 9 medidas diretas
 - Área
 - Perímetro
 - Diâmetro
 - Altura Fisiológica
 - Largura fisiológica
 - 4 Áreas parciais

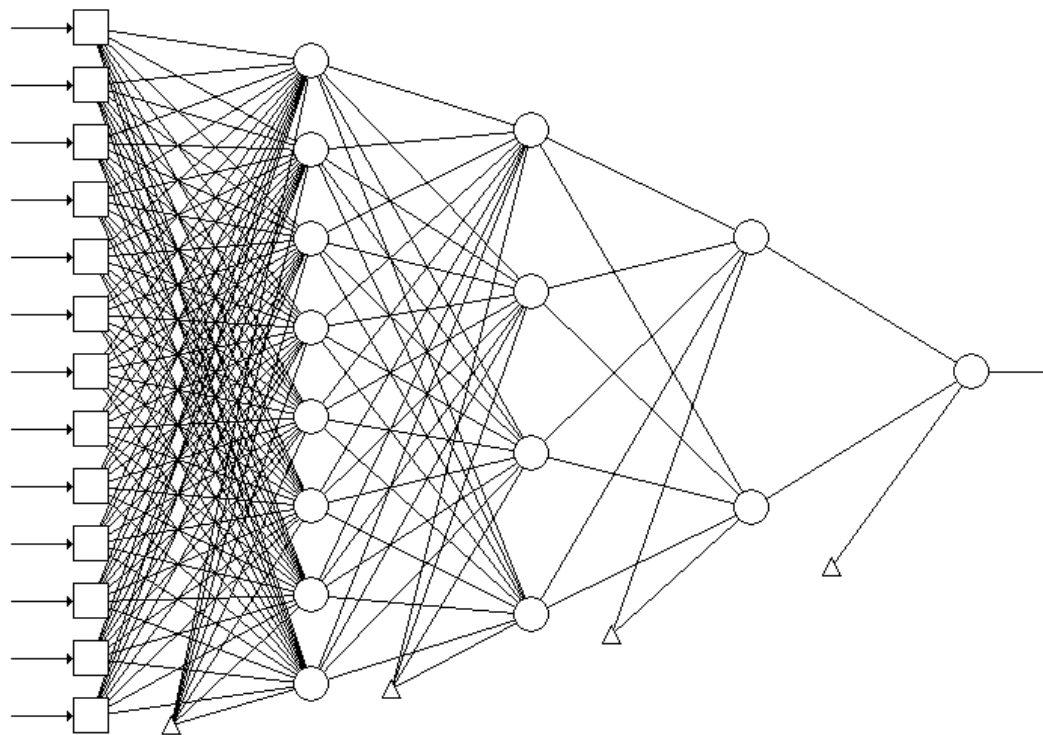


4 Áreas



A rede neural

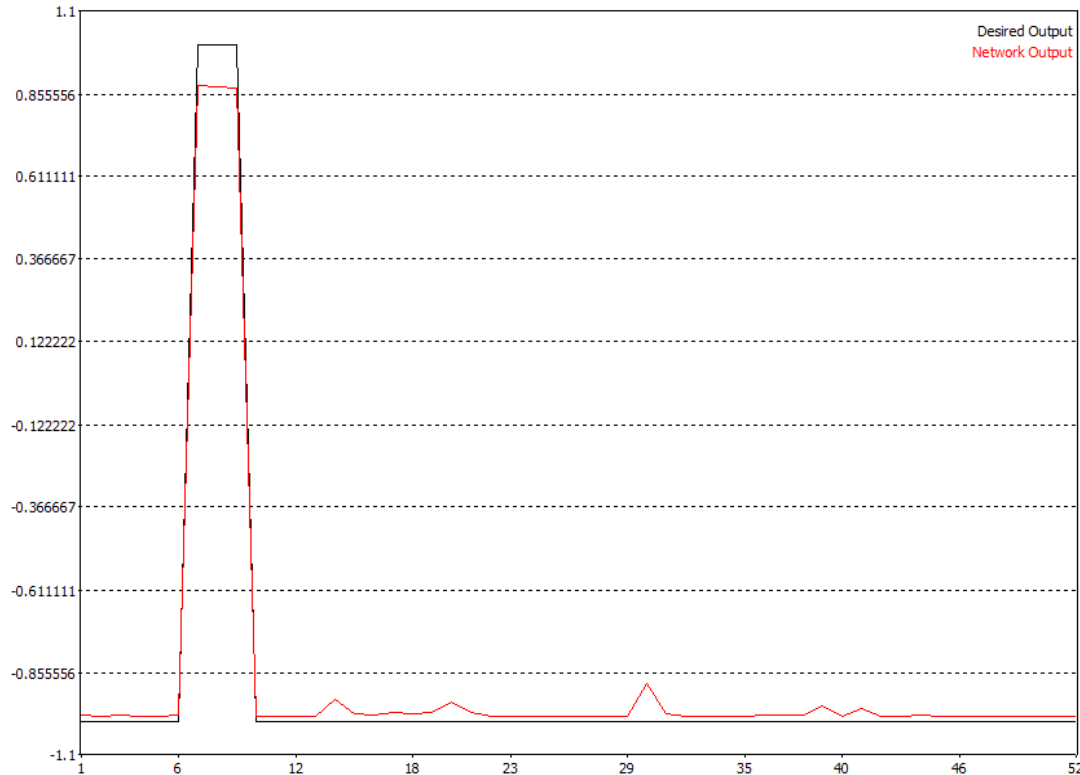
13-8-4-2-1



Primeira
hidden layer
com 8 a 11
neurônios

Resultados

I

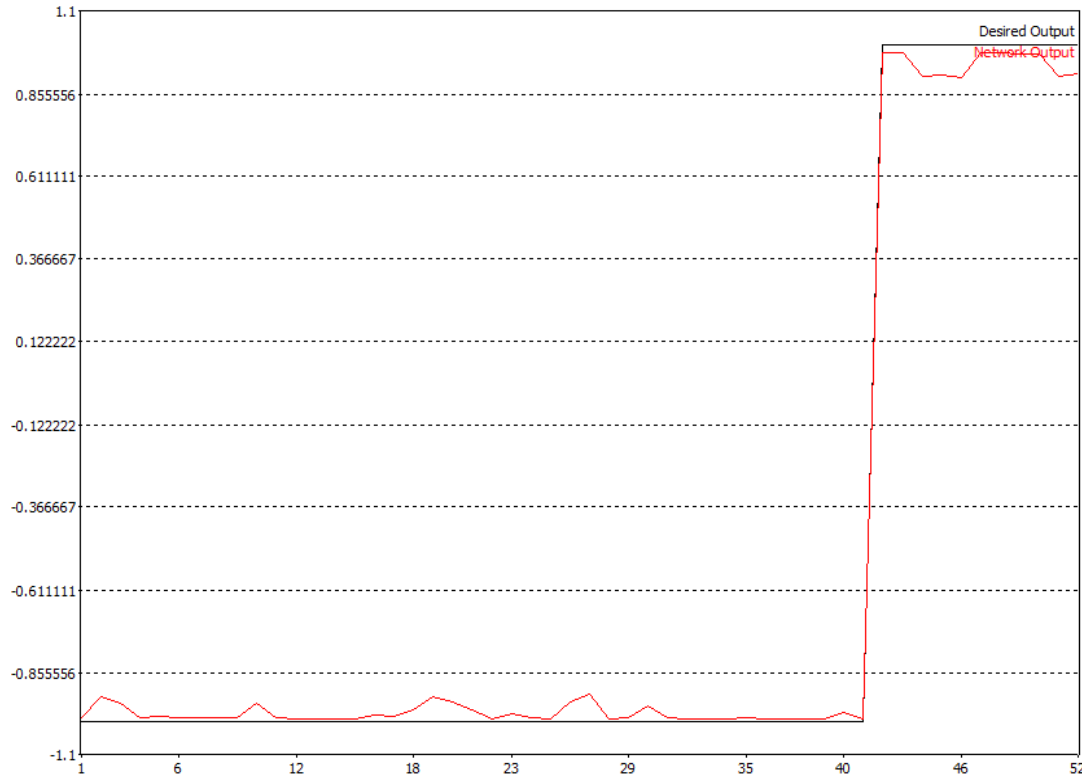


100% de
acertos com
limiar em 0



Resultados

II

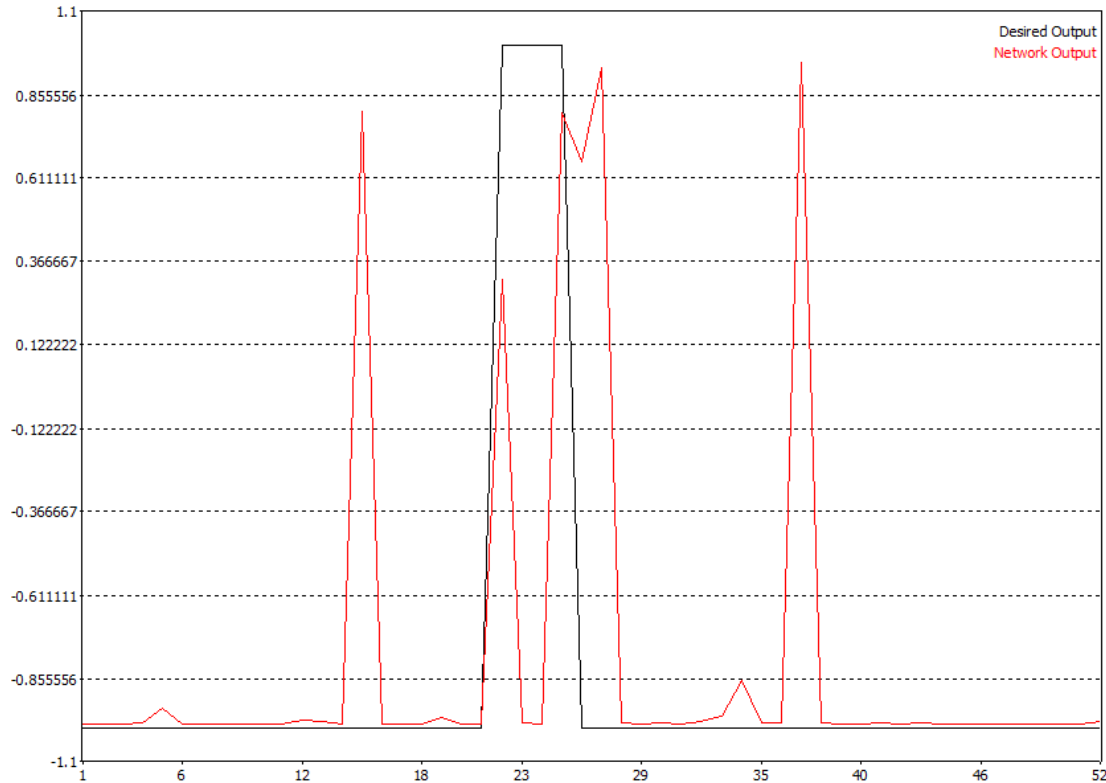


100% de
acertos com
limiar em 0



Resultados

IV



Folha tipo 6:
Difícil Reconhecimento



- Reconhecimento 100% efetivo para 2 tipos de folha.
- Taxa de acerto entre 70% e 95% para outros 8 tipos.
- Taxa de acerto muito baixa ($<70\%$) para 2 tipos de folha.

Refinamento

Randomização de dados em conjuntos de treinamento e teste:

- loop para obter o melhor mínimo local
- Aumenta a complexidade.
- Loop para fazer k-fold cross validation

Refinamento

- Redes de 1 a 50 neurônios em uma camada escondida.
- Sem falsos positivos, apenas falsos negativos
- Taxa de acerto $>90\%$

Discussão

- As folhas com maior taxa de acerto eram também folhas com maior número de amostras (salvo 1 exceção).
- Resultados negativos
 - Limitação dos aspectos geométricos?
 - Necessidade de N amostral maior.

Conclusão

- Apenas aspectos geométricos já são suficientes para o reconhecimento de uma grande quantidade de tipos de folha.
- Sem o aumento da quantidade de parâmetros, tão somente pelo aumento do N amostral, pode ser possível aperfeiçoar a rede até onde se queira.