

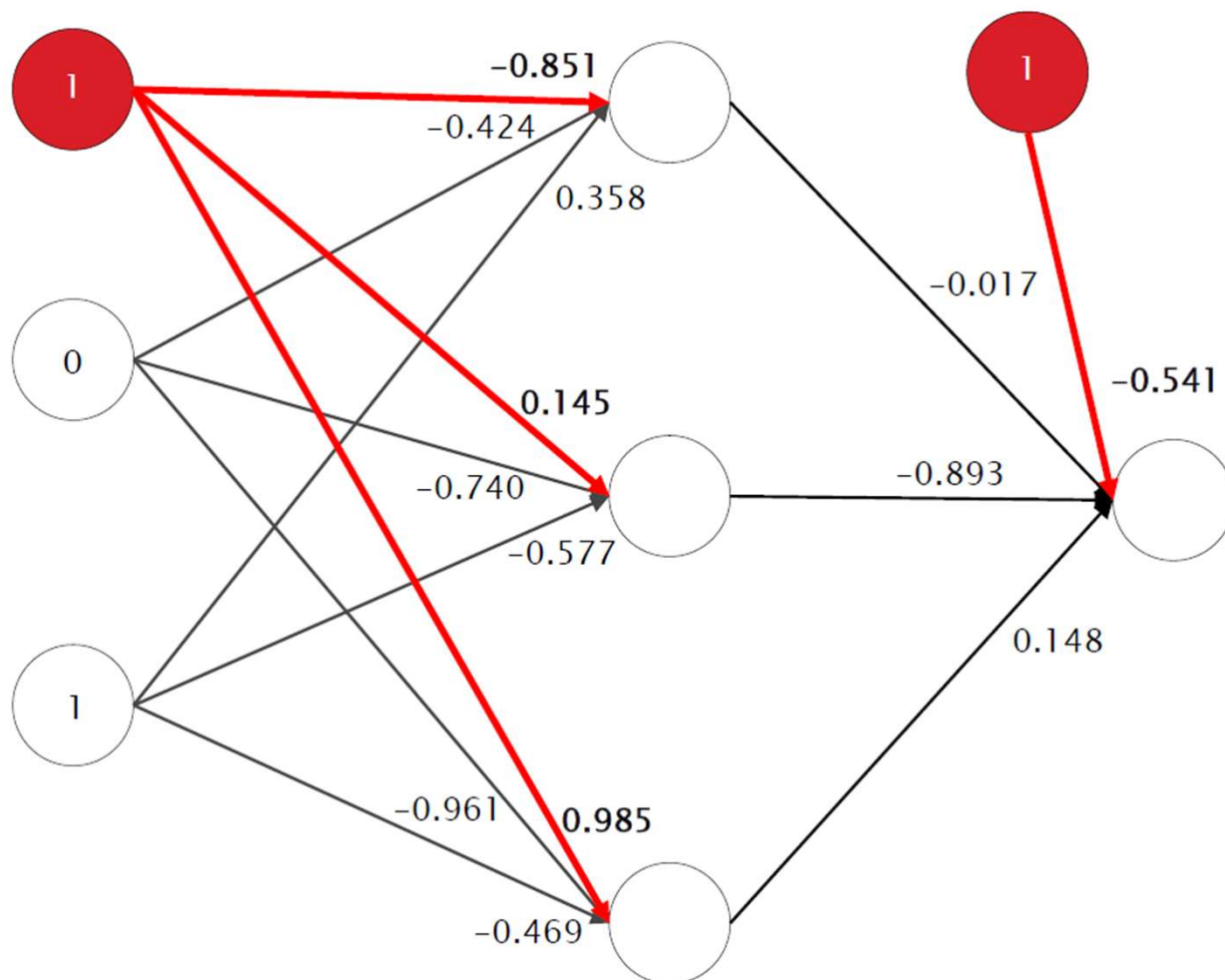
Redes Neurais Multicamadas

Prof. Gustavo Willam Pereira



INSTITUTO FEDERAL
Sudeste de Minas Gerais

Bias (Viés)



Bias (Viés)

- Valores diferentes mesmo se todas as entradas forem zero
- Muda a saída com a unidade de bias

Erro

- Algoritmo mais simples

$$\text{erro} = \text{respostaCorreta} - \text{respostaCalculada}$$

X1	X2	CLASSE	CALCULADO	ERRO
0	0	0	0.406	-0.406
0	1	1	0.432	0.568
1	0	1	0.437	0.563
1	1	0	0.458	-0.458

Média absoluta = 0.49

Outros cálculos dos erros

- MSE (Mean square error)

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2$$

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (f_i - o_i)^2}$$

Outros cálculos dos erros

x1	x2	Classe	Calculado	Erro
0	0	0	0.406	$(0 - 0.406)^2 = 0.164$
0	1	1	0.432	$(1 - 0.432)^2 = 0.322$
1	0	1	0.437	$(1 - 0.437)^2 = 0.316$
1	1	0	0.458	$(0 - 0.458)^2 = 0.209$

Soma = 1.011

$MSE = 1.011 / 4 = 0.252$

$RMSE = 0.501$

Saídas com mais neurônios

- Problemas mais complexos

Pode ser necessário colocar mais de um neurônio na camada de saída

Será necessário realizar uma codificação para determinar a saída

Ex: Análise de Risco (Saída: 2 Neurônios)

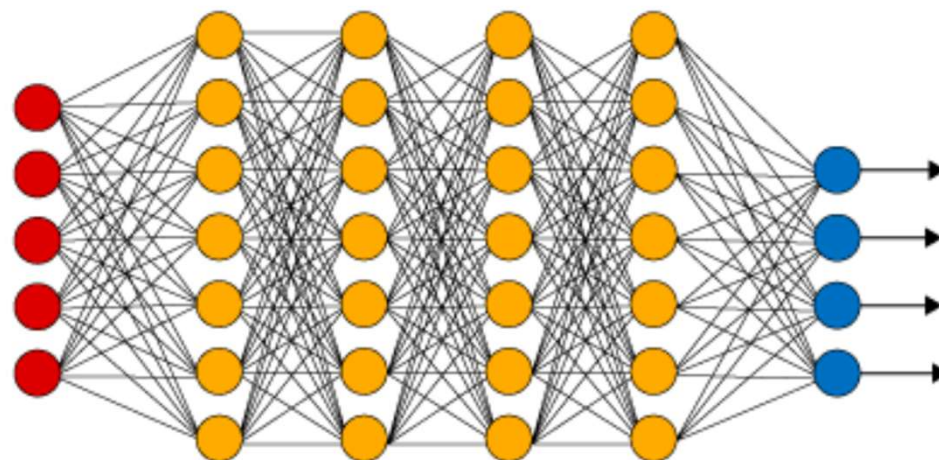
Classe Baixa 0 – 0

Classe Média 0 – 1

Classe Alta 1 – 1

Deep Learning

- Anos 90: Máquinas de vetores de suporte (SVM)
- A partir de 2006 foram criados algoritmos para treinamento de redes neurais
- Duas ou mais camadas escondidas (redes profundas)



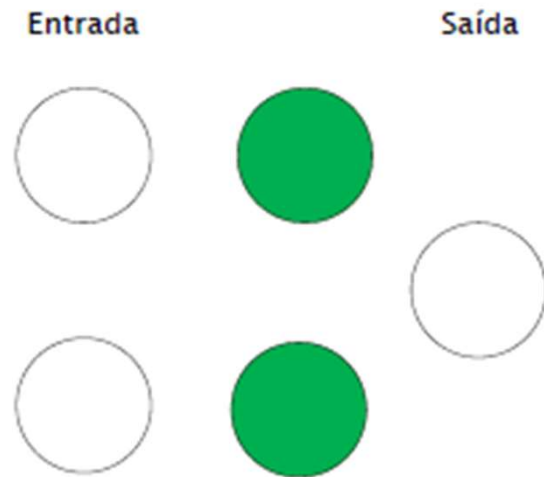
Deep Learning

- São utilizadas outras técnicas
- “Problema do gradiente desaparecendo” –vanishing gradient problem–gradiente fica muito pequeno, mudanças nos pesos ficam pequenas
- Outras funções de ativação
- Redes neurais convolucionais
- Redes neurais recorrentes
- Keras, TensorFlow
- FastIA, PyTorch
- GPU

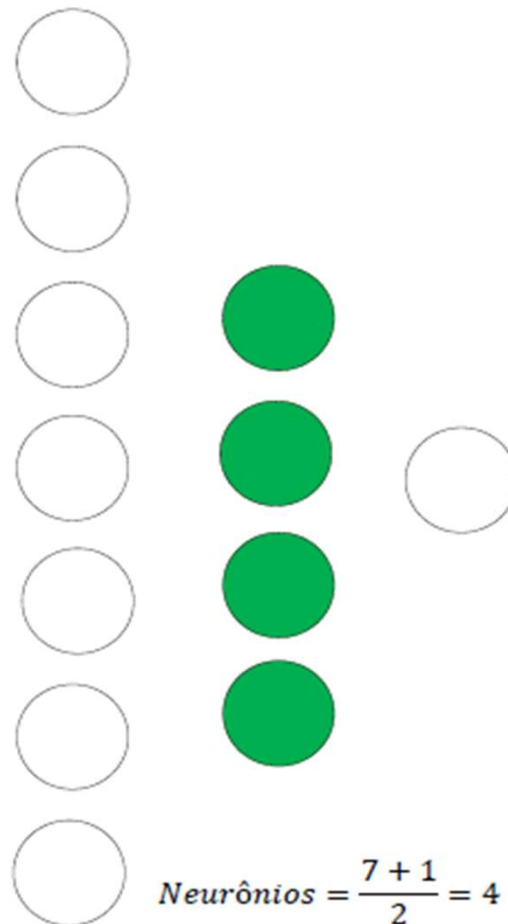
Camadas Ocultas

- Quantidade de Neurônios na Camada Oculta

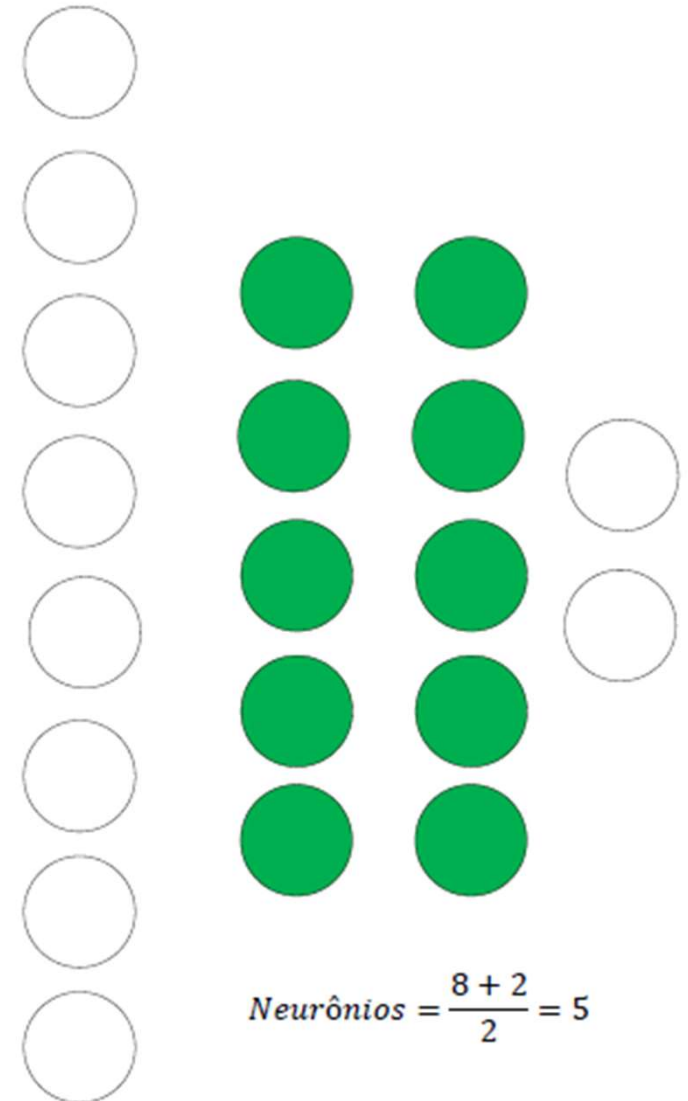
$$\text{Neurônios} = \frac{\text{Entradas} + \text{Saídas}}{2}$$



$$\text{Neurônios} = \frac{2 + 1}{2} = 1.5$$



$$\text{Neurônios} = \frac{7 + 1}{2} = 4$$



$$\text{Neurônios} = \frac{8 + 2}{2} = 5$$

Camadas Ocultas

- Problemas linearmente separáveis não necessitam de camadas ocultas
- Cross validation(validação cruzada)
- Em geral, duas camadas funcionam bem para poucos dados
- Pesquisas em deep learning mostram que mais camadas são essenciais para problemas mais complexos

Exemplo de um DataSet para avaliar risco de Cliente

História do crédito	Dívida	Garantias	Renda anual	Risco
Ruim	Alta	Nenhuma	< 15.000	Alto
Desconhecida	Alta	Nenhuma	>= 15.000 a <= 35.000	Alto
Desconhecida	Baixa	Nenhuma	>= 15.000 a <= 35.000	Moderado
Desconhecida	Baixa	Nenhuma	> 35.000	Alto
Desconhecida	Baixa	Nenhuma	> 35.000	Baixo
Desconhecida	Baixa	Adequada	> 35.000	Baixo
Ruim	Baixa	Nenhuma	< 15.000	Alto
Ruim	Baixa	Adequada	> 35.000	Moderado
Boa	Baixa	Nenhuma	> 35.000	Baixo
Boa	Alta	Adequada	> 35.000	Baixo
Boa	Alta	Nenhuma	< 15.000	Alto
Boa	Alta	Nenhuma	>= 15.000 a <= 35.000	Moderado
Boa	Alta	Nenhuma	> 35.0000	Baixo
Ruim	Alta	Nenhuma	>= 15.000 a <= 35.000	Alto

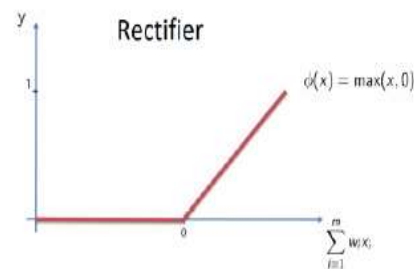
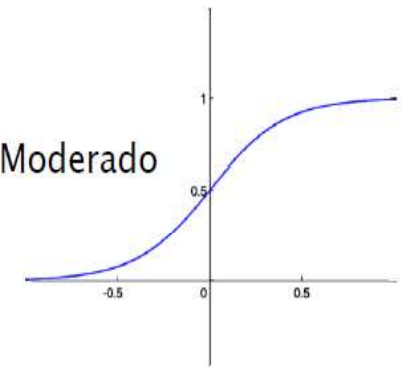
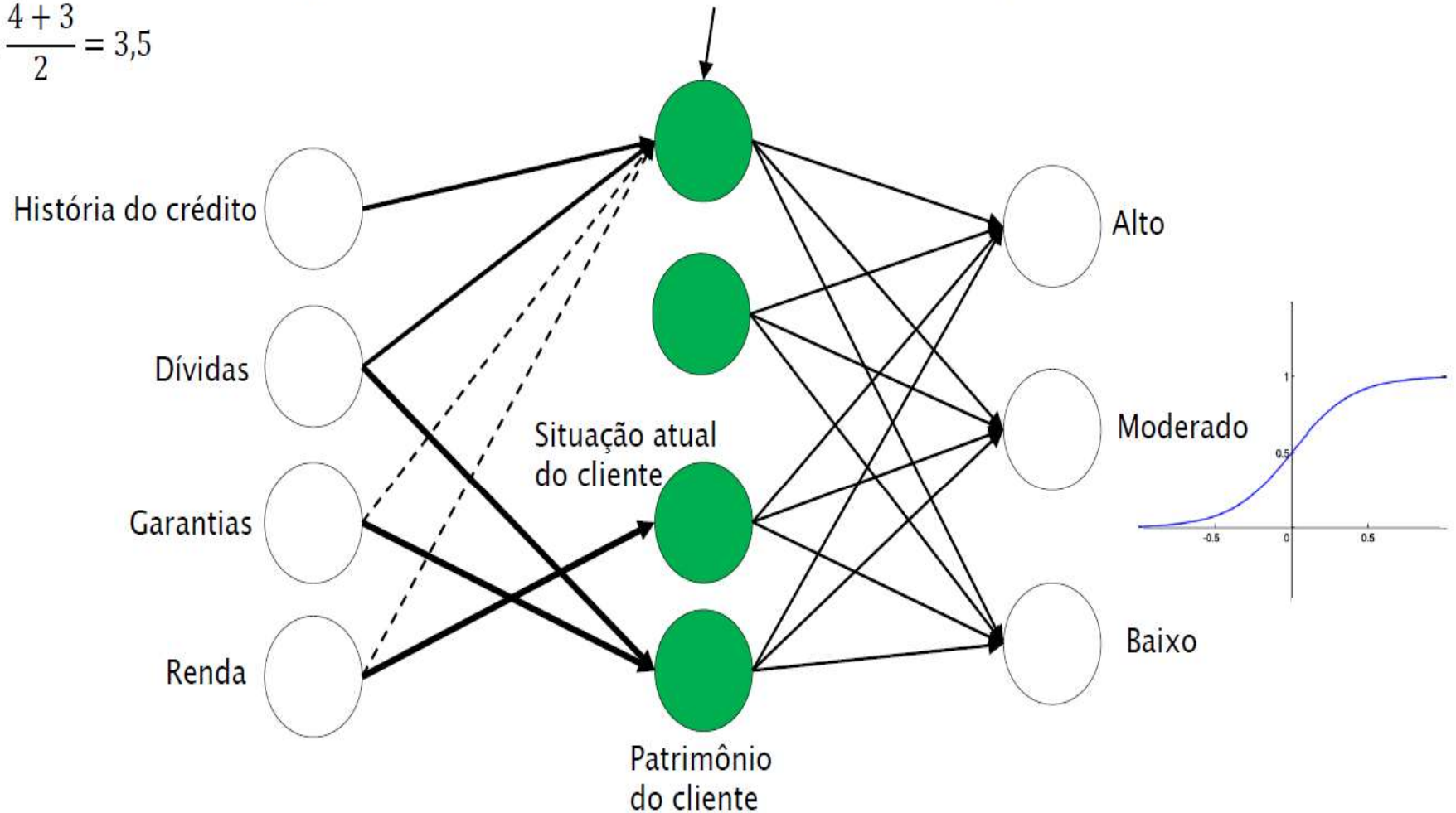
Exemplo de um DataSet para avaliar risco de Cliente

História do crédito	Dívida	Garantias	Renda anual	Risco
3	1	1	1	Alto
2	1	1	2	Alto
2	2	1	2	Moderado
2	2	1	3	Alto
2	2	1	3	Baixo
2	2	2	3	Baixo
3	2	1	1	Alto
3	2	2	3	Moderado
1	2	1	3	Baixo
1	1	2	3	Baixo
1	1	1	1	Alto
1	1	1	2	Moderado
1	1	1	3	Baixo
3	1	1	2	Alto



Procurando pelo histórico da pessoa
Dispara quando os dois critérios ocorrem, contribuindo para o risco

$$\text{Neurônios} = \frac{4 + 3}{2} = 3,5$$



Quanto maior o valor da ativação,
mais impacto o neurônio possui

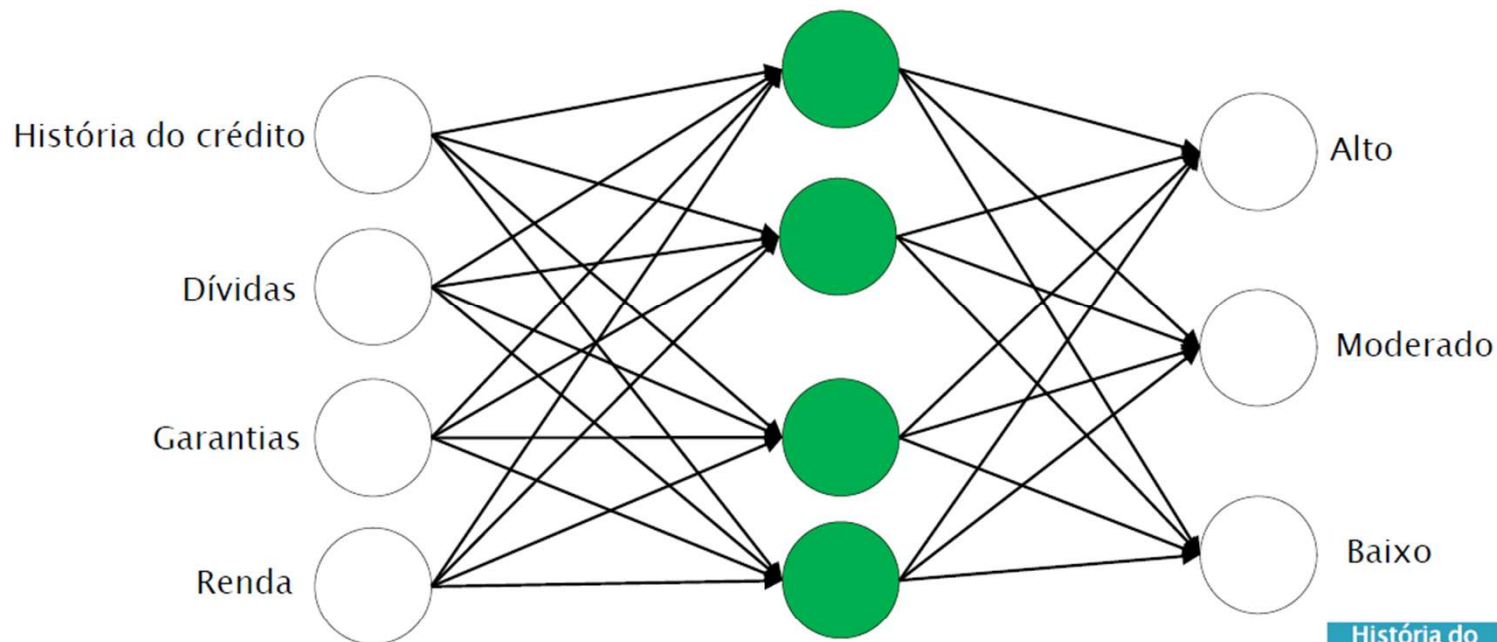
Camada de saída categórica

- Realizar um encoder para cada saída:

Ex: Alto = 1 0 0

Moderado = 0 1 0

Baixo = 0 0 1



$\text{erro} = \text{respostaCorreta} - \text{respostaCalculada}$

$\text{respostaCorreta} = 1\ 0\ 0$

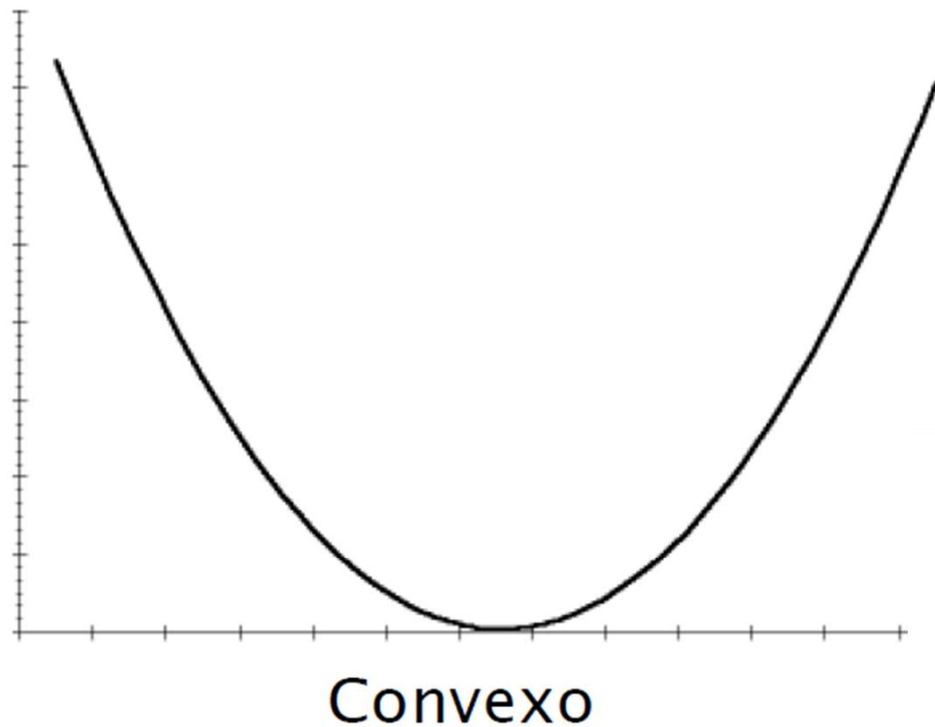
$\text{respostaCalculada} = 0.98\ 0.12\ 0.05$

$\text{erro} = (1 - 0.98) + (0 - 0.12) + (0 - 0.05)$

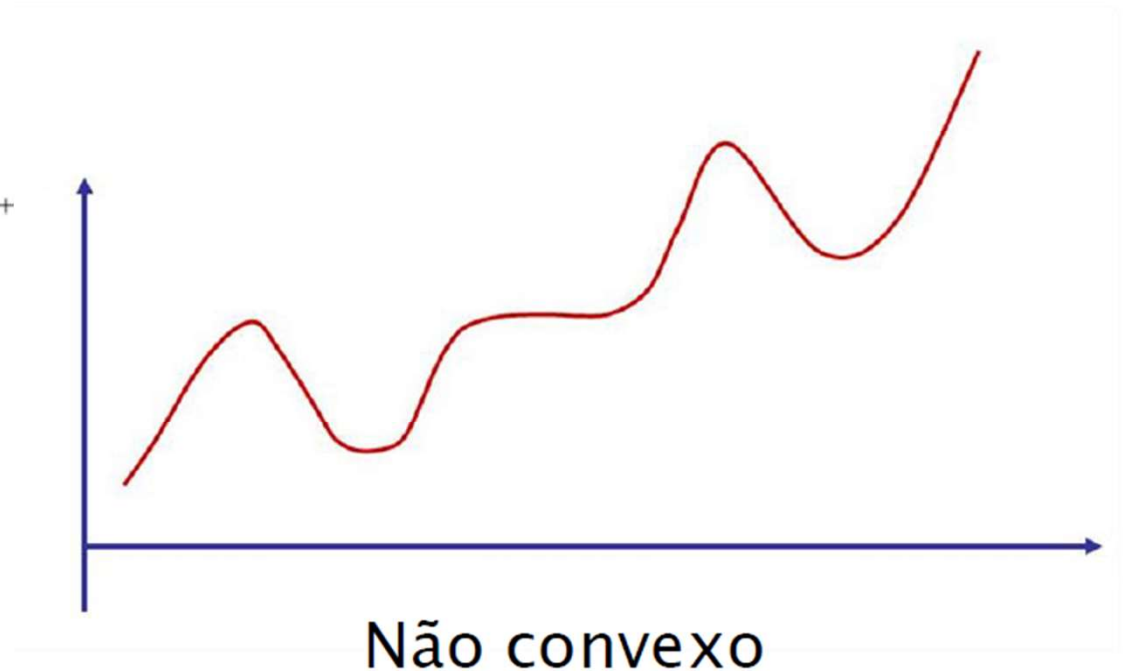
$\text{erro} = 0.02 + 0.12 + 0.05 = 0.19$

História do crédito	Dívida	Garantias	Renda anual	Risco
3	1	1	1	100
2	1	1	2	100
2	2	1	2	010
2	2	1	3	100
2	2	1	3	001
2	2	2	3	001
3	2	1	1	100

Descida do gradiente estocástico



- Problema com mínimos locais
- Em superfícies convexas existe apenas o mínimo global
- Em superfícies não convexas existe mínimos locais e mínimo global



Descida do gradiente estocástico

História do crédito	Dívida	Garantias	Renda anual	Risco
3	1	1	1	100
2	1	1	2	100
2	2	1	2	010
2	2	1	3	100
2	2	1	3	001
2	2	2	3	001
3	2	1	1	100
3	2	2	3	010
1	2	1	3	001
1	1	2	3	001
1	1	1	1	100
1	1	1	2	010
1	1	1	3	001
3	1	1	2	100

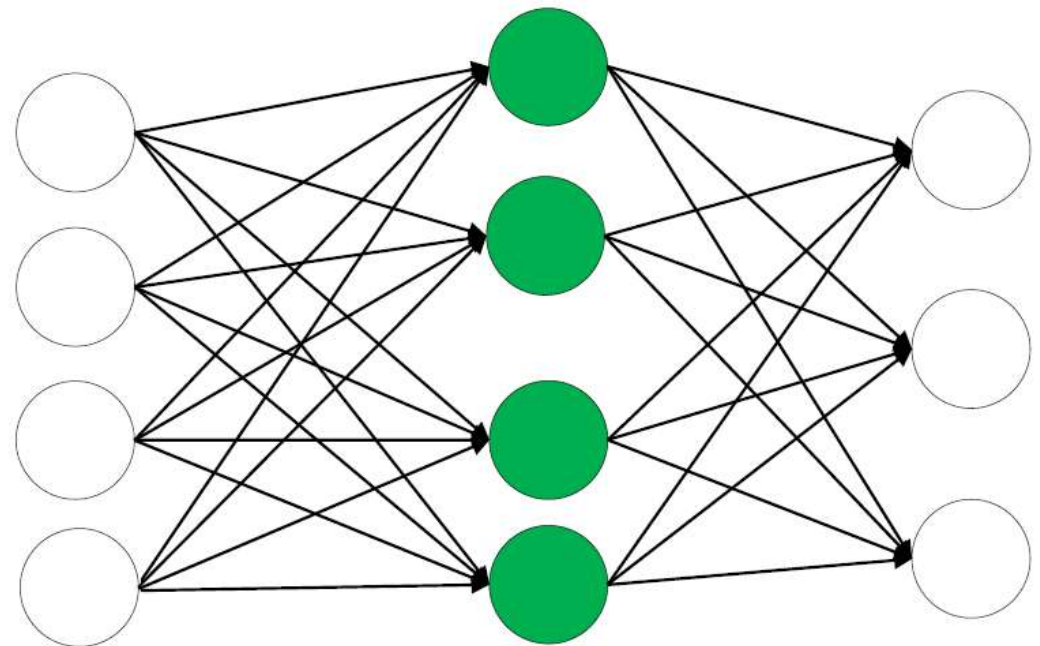
Calcula o erro para todos os registros e atualiza os pesos

Batch gradient descent

Calcula o erro para cada registro e atualiza os pesos

Stochastic gradient descent

História do crédito	Dívida	Garantias	Renda anual	Risco
3	1	1	1	100
2	1	1	2	100
2	2	1	2	010
2	2	1	3	100
2	2	1	3	001
2	2	2	3	001
3	2	1	1	100
3	2	2	3	010
1	2	1	3	001
1	1	2	3	001
1	1	1	1	100
1	1	1	2	010
1	1	1	3	001
3	1	1	2	100



Gradiente Descendente

- Stochastic

Ajuda a prevenir mínimos locais (superfícies não convexas)

Mais rápido (não precisa carregar todos os dados em memória)

- Mini batch gradiente descent

Escolhe um número de registros para rodar e atualizar os pesos

Implementação de Redes Neurais Multicamadas

- Exemplo: Breast Cancer Wisconsin
- Redes Neurais com Sklearn (dataset Iris)

Considerações finais

- Aplicações de redes neurais
- Perceptron simples de uma camada - passo a passo (teoria e prática)
- Redes neurais multicamada

Ajuste dos pesos

Delta

Gradient descente (descida do gradiente)

Backpropagation (retropagação)

Implementação passo a passo

- Redes Neurais com scikit-learn



INSTITUTO FEDERAL
Sudeste de Minas Gerais