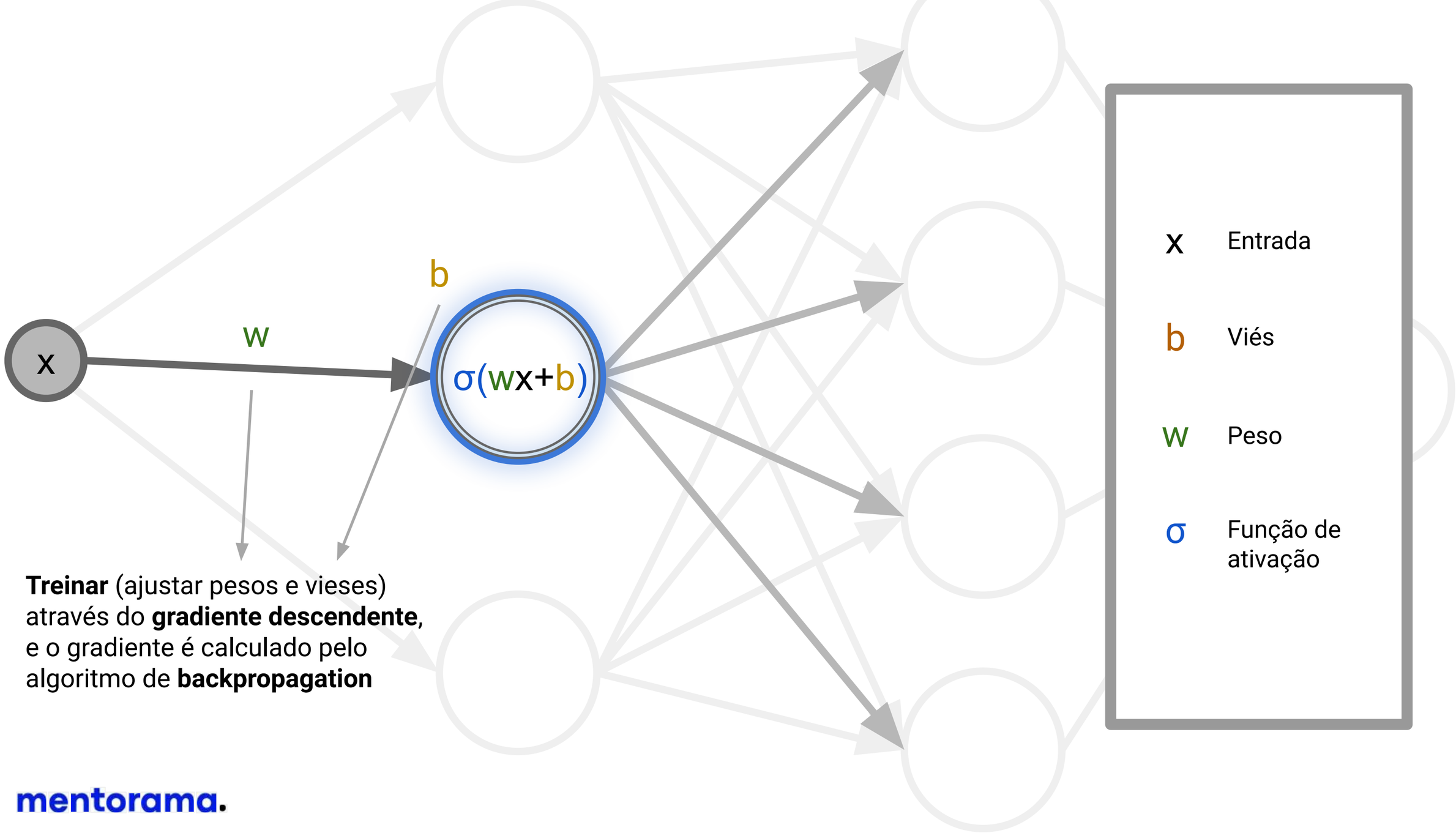
Capítulo 3

mentorama.

mentorama.

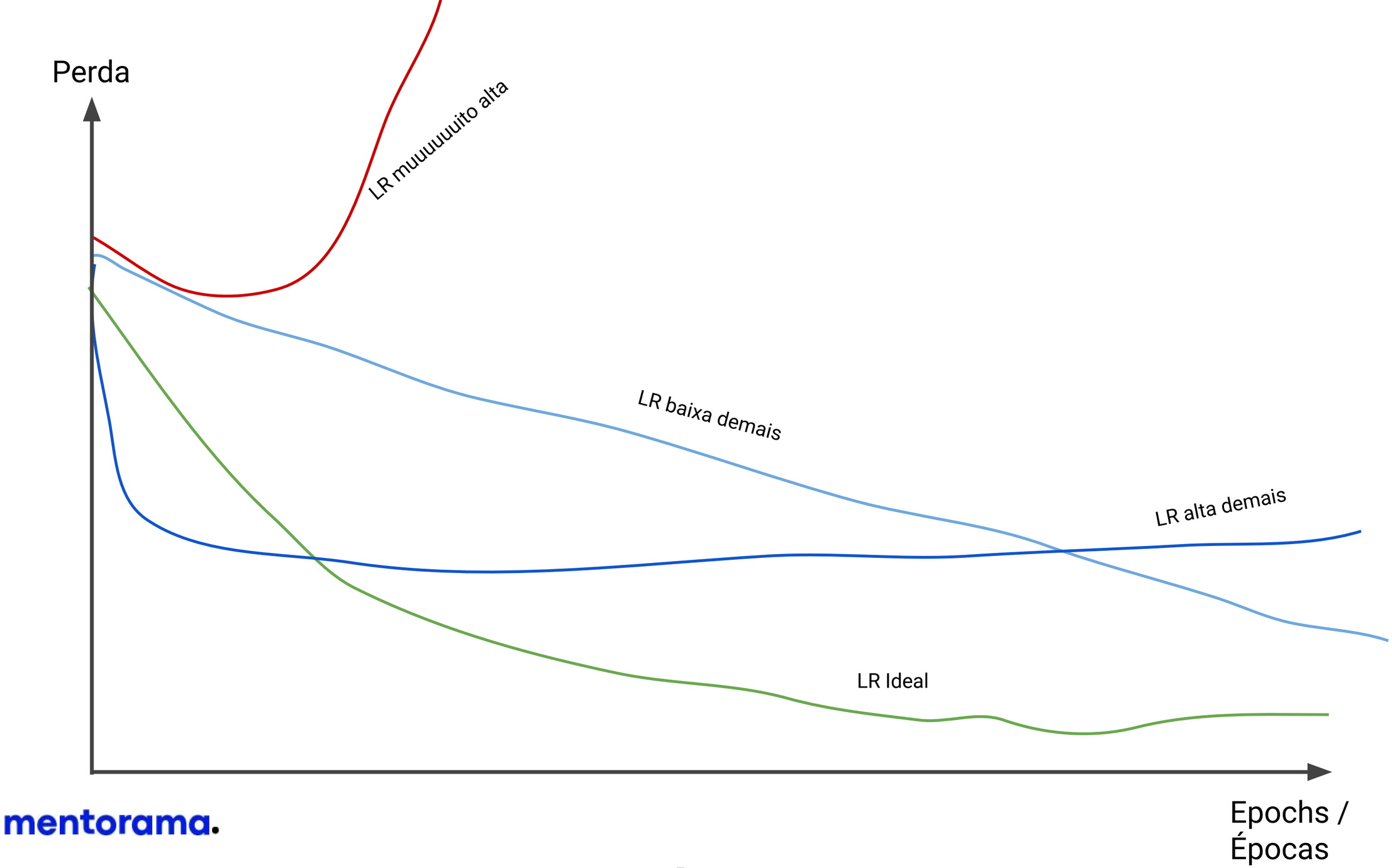


Treinar (ajustar pesos e vieses) através do **gradiente descendente**, e o gradiente é calculado pelo algoritmo de **backpropagation**

x	Entrada
b	Viés
w	Peso
σ	Função de ativação

mentorama.

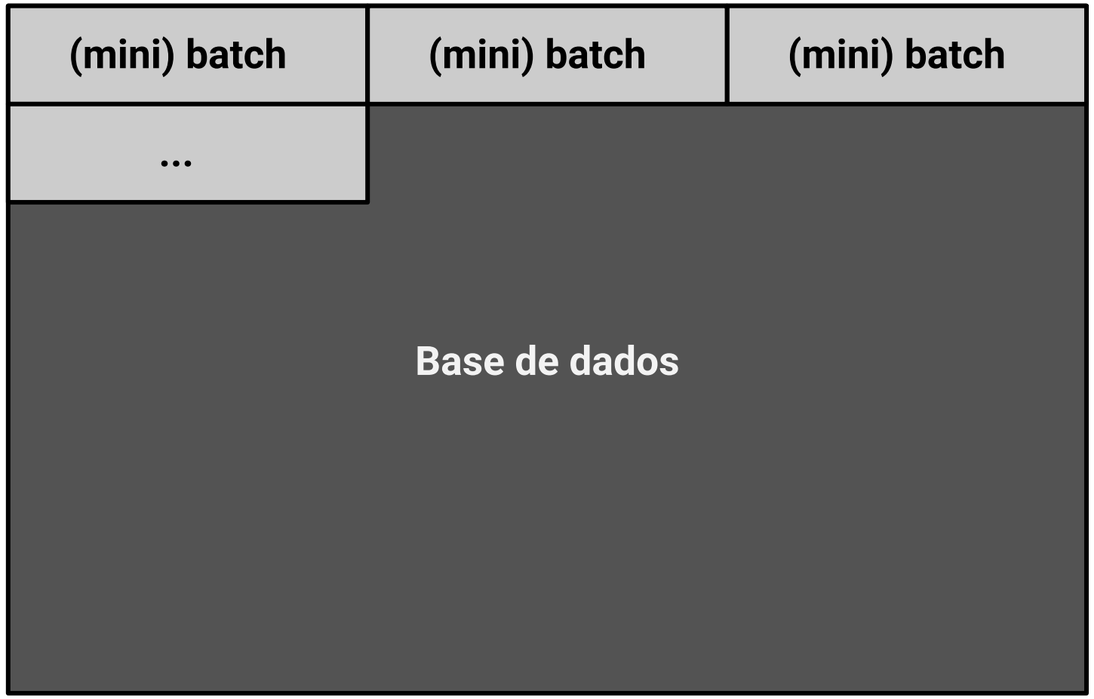
mentorama.



mentorama.

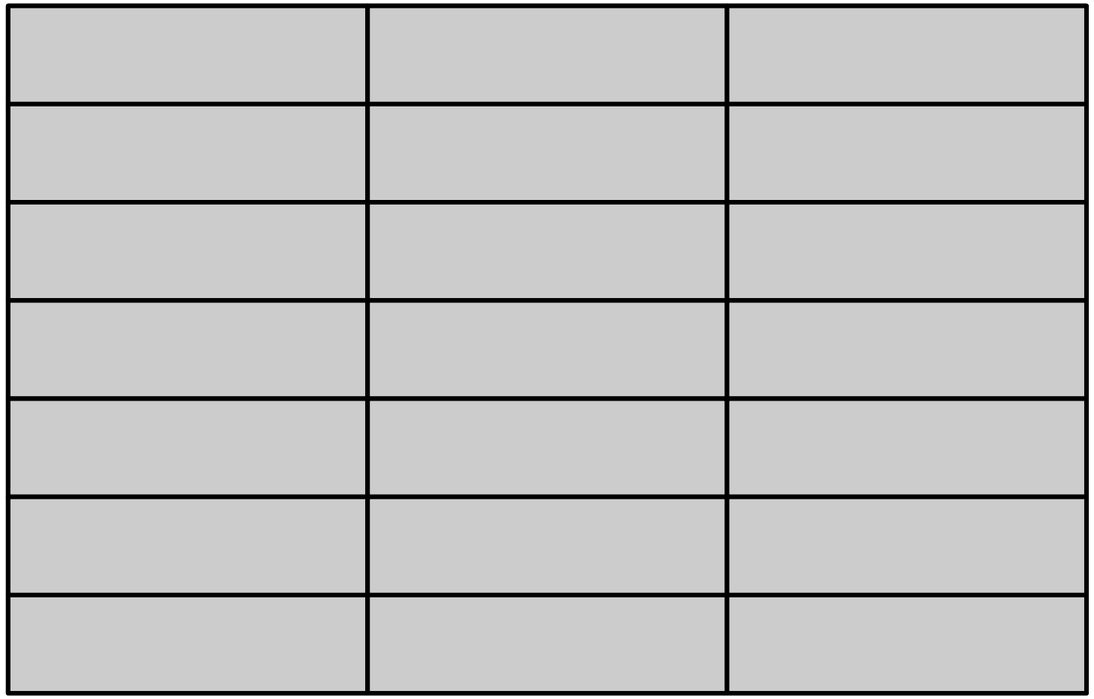
mentorama.

mentorama.



mentorama.

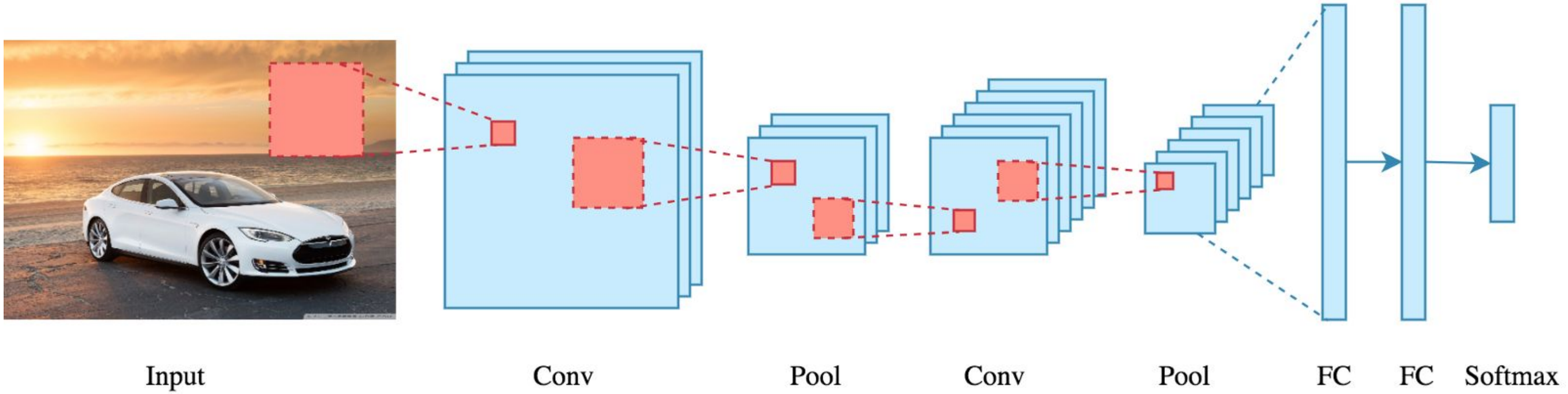
1 epoch



mentorama.

Outros tipos de redes:

Convolucionais (CNNs, Convolutional Neural Networks)



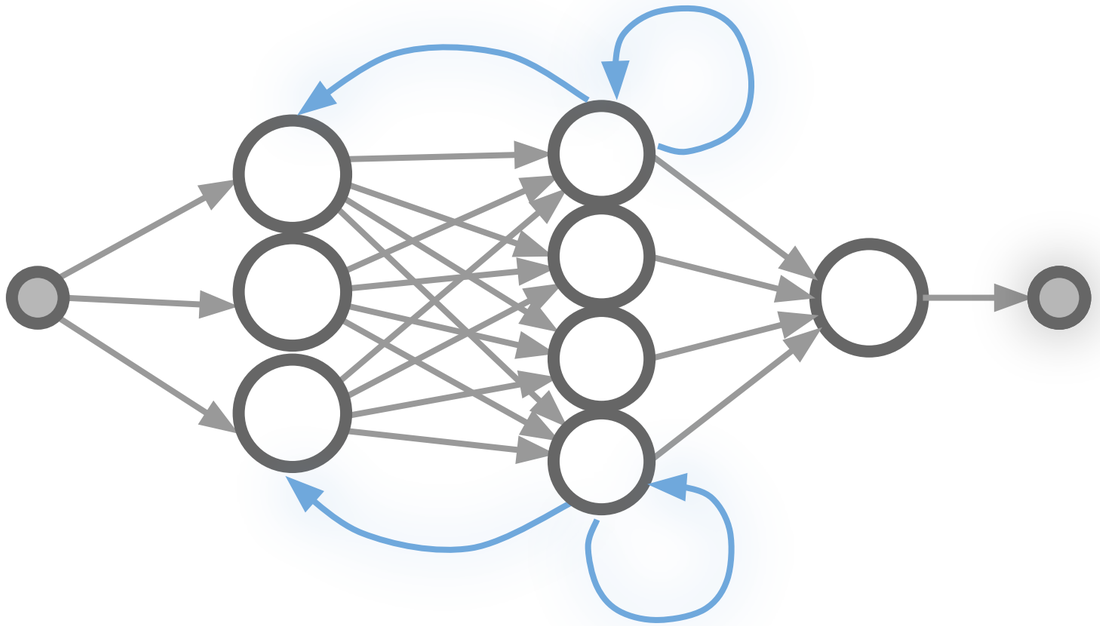
Definidas pelas camadas com operações de convolução.

Muito boas para lidar com imagens!

Fonte da imagem: https://nitinpanwar98.medium.com/convolutional-neural-networks-cnn-explanation-and-implementation-8c41be8b51a

Outros tipos de redes:

Recorrentes (RNNs, Recurrent Neural Networks)

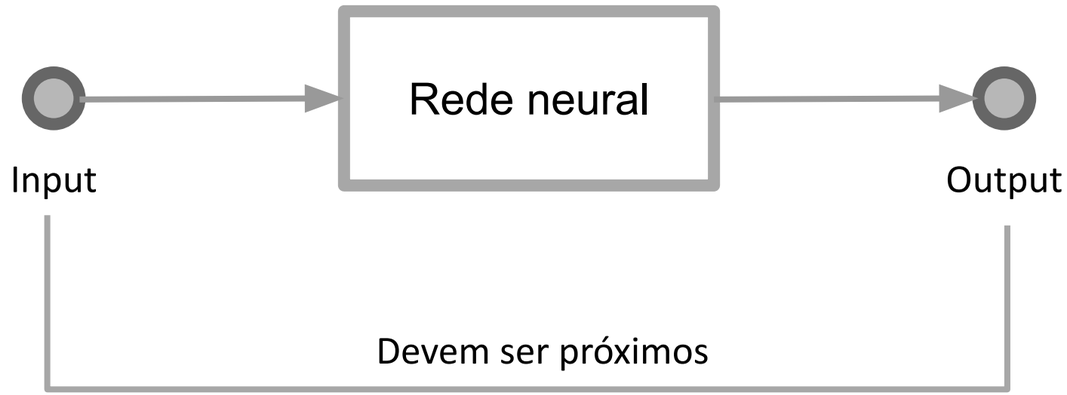


Quebram a regra de ser completamente "feedforward", adicionando ligações para neurônios de camadas anteriores ou até da mesma camada.

São ótimas para dados com algum tipo de dependência sequencial.

Outros tipos de redes:

Autoencoders

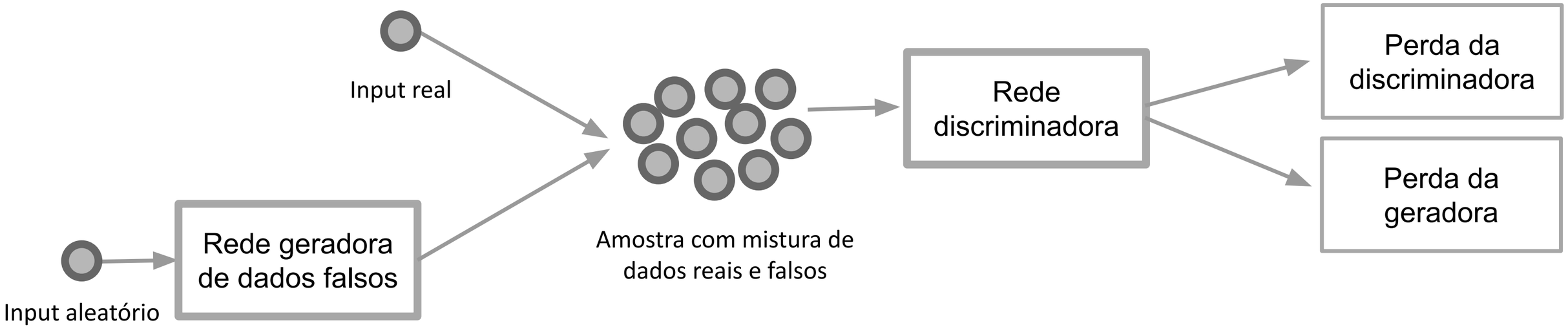


A ideia mais básica é ter uma rede que gera dados iguais aos que você apresenta para ela - o que importa é que ela aprenda uma boa representação dos seus dados

Também são usadas para retirar ruído de dados

Outros tipos de redes:

Generativas adversariais (GANs, Generative Adversarial Networks)



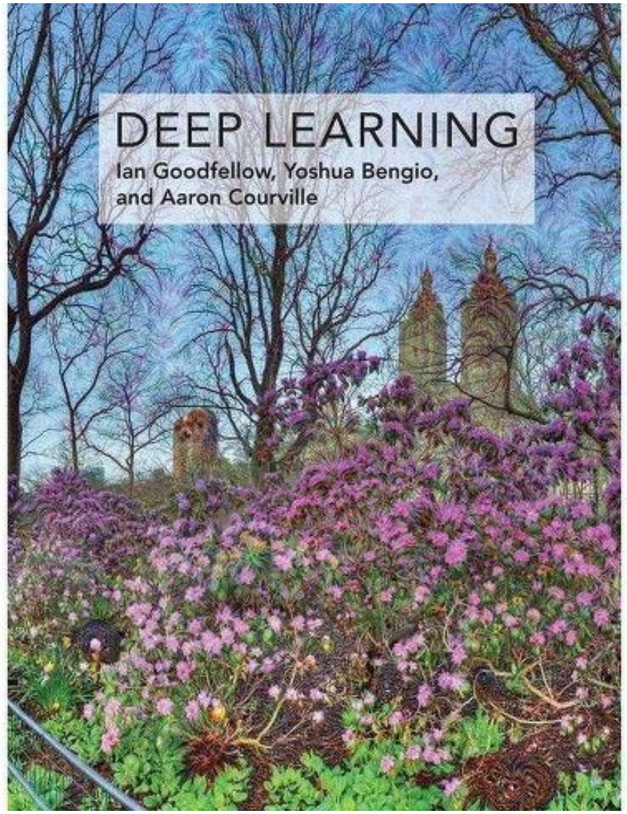
São duas redes trabalhando juntas, uma que gera dados falsos e outra que discrimina entre dados falsos e dados reais

8.5.4 Choosing the Right Optimization Algorithm

We have discussed a series of related algorithms that each seek to address the challenge of optimizing deep models by adapting the learning rate for each model parameter. At this point, a natural question is: which algorithm should one choose?

Unfortunately, there is currently no consensus on this point. Schaul et al. (2014) presented a valuable comparison of a large number of optimization algorithms across a wide range of learning tasks. While the results suggest that the family of algorithms with adaptive learning rates (represented by RMSProp and AdaDelta) performed fairly robustly, no single best algorithm has emerged.

Currently, the most popular optimization algorithms actively in use include SGD, SGD with momentum, RMSProp, RMSProp with momentum, AdaDelta, and Adam. The choice of which algorithm to use, at this point, seems to depend largely on the user's familiarity with the algorithm (for ease of hyperparameter tuning).

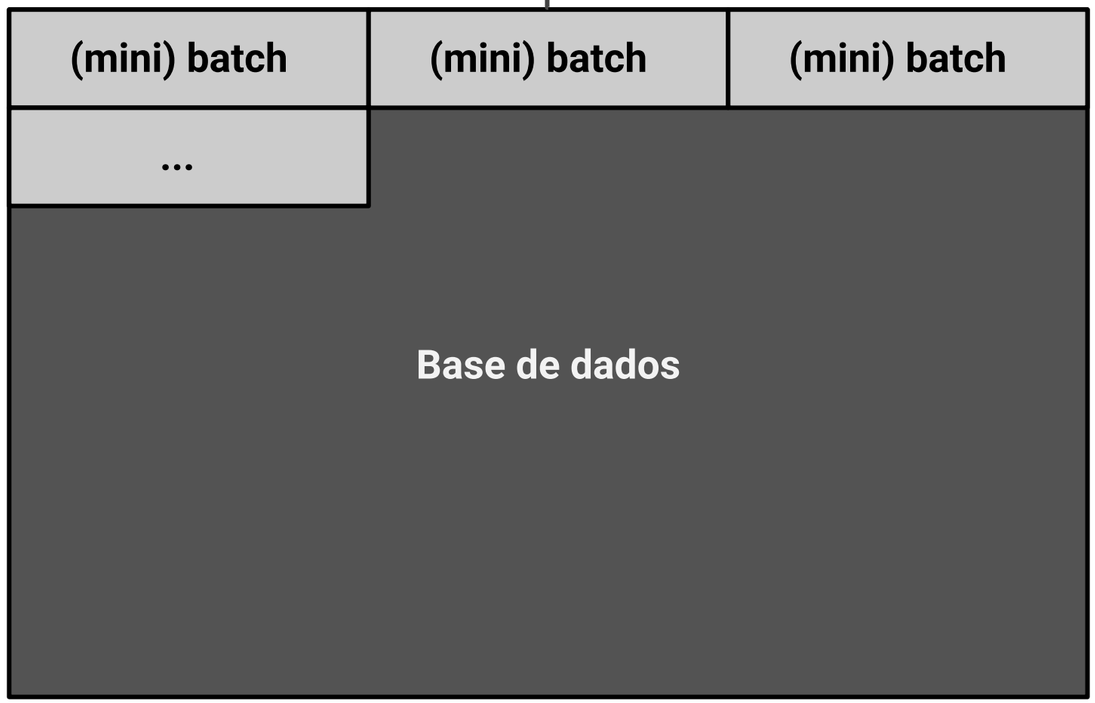


Ian Goodfellow, Yoshua Bengio and Aaron Courville - *Deep Learning,* MIT Press, 2016

Lembrança: O desvio padrão é a raiz quadrada da variância!

Batch normalization *(Sergey Ioffe e Christian Szegedy, 2015)*

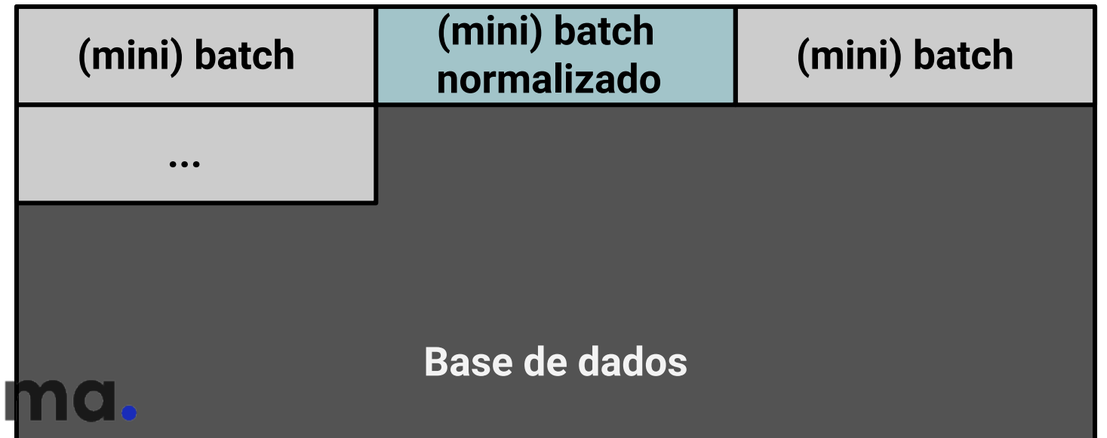
Calcular, para este batch:
m(x), a média, e **DP(x)**, o desvio padrão



Depois, padronizar o batch com esses valores:

$$\text{(mini) batch normalizado} = \frac{\text{(mini) batch} - m(\text{(mini) batch})}{DP(\text{(mini) batch})}$$

Isso é feito para todos os batches.



mentorama.

Lista dos otimizadores em https://keras.io/api/optimizers/ :

Algoritmo	Descrição
SGD	Gradiente descendente com ou sem momento / momento de Nesterov
RMSprop	Mantém uma média histórica do quadrado dos gradientes, e divide pela raiz desse valor
Adam	Calcula o primeiro e segundo momento do gradiente e faz uma "normalização" com esses valores
Adadelta	Versão mais robusta do Adagrad
Adagrad	A learning rate ("tamanho do passo") vai ser individual para cada parâmetro, relativa ao quão frequentemente esse parâmetro é atualizado durante o treinamento
Adamax	Versão do Adam que usa a norma infinita, também chamada de norma max
Nadam	Adam com momento de Nesterov
Ftrl	Faz SGD com regularização L1 e L2

Referência *sensacional* com GIFs:
https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c

Grid search

1. Dentro dos valores possíveis para um parâmetro, escolha alguns. (Não precisam ser igualmente espaçados, é até interessante que tenha valores em escalas bem discrepantes, para que você possa explorar várias ordens de grandeza inicialmente.)
   Ex: {-2, -1, 0, 1, 2}

2. Rode o código com cada um desses valores possíveis e monitore alguma métrica de interesse.
   Ex: Função perda

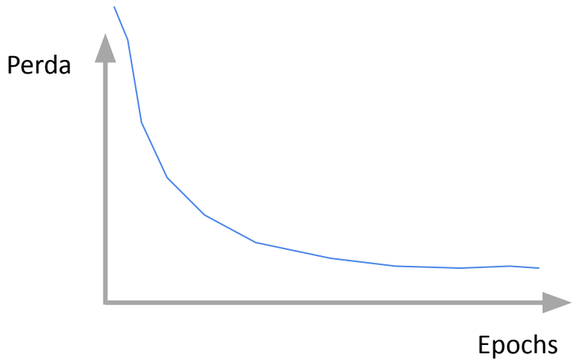
3. Escolha o valor do parâmetro que dá a melhor métrica de interesse.
   Ex: Rodou para esses valores e viu que a menor perda foi observada quando o valor do parâmetro era 1

4. Repita a procura explorando mais valores próximos desse valor, ou decida por esse valor e encerre a procura
   Ex: Repetir a procura em {0.5, 0.75, 1, 1.25, 1.5} ou decidir usar 1

mentorama.

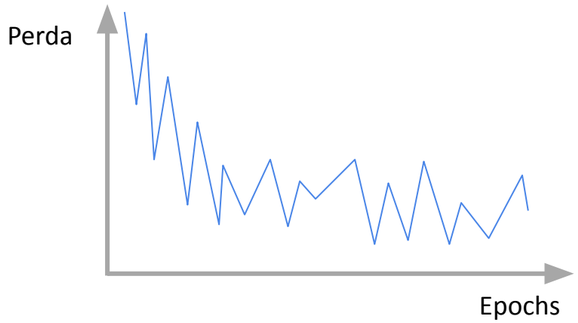
Maior batch size

- Mais acurácia na estimativa do gradiente que vai ser usado no algoritmo de gradiente descendente
- O gráfico da perda ao longo das epochs fica mais estável no seu decrescimento, porque o caminho feito pelo gradiente descendente fica menos errático
- No entanto, cada estimativa fica mais demorada e isso atrasa o treinamento



Menor batch size

- O gradiente a ser usado no algoritmo de gradiente descendente vai ser estimado com menor acurácia
- O gráfico da perda ao longo das epochs fica mais errático, então pode ficar difícil de ver se o aprendizado estabilizou
- Agiliza o treinamento já que cada iteração é mais rápida - ótimo para explorar grandes partes do espaço paramétrico



Regularização

A ideia é adicionar uma restrição aos parâmetros da rede (w, b) no processo de minimização da função custo.

Sem regularização: vamos tentar encontrar os pesos e vieses que minimizem a função perda
Com regularização: vamos tentar encontrar os pesos e vieses que minimizem a função perda, mas esses pesos e vieses não podem ser muito grandes também!

Regularização L1
(para os pesos)

$$L(\hat{y}, y) + \lambda \sum_{i=1}^{p} |w_i|$$

Esse é o termo de regularização para os pesos: queremos minimizar também, além da função custo, a soma absoluta dos p pesos da rede

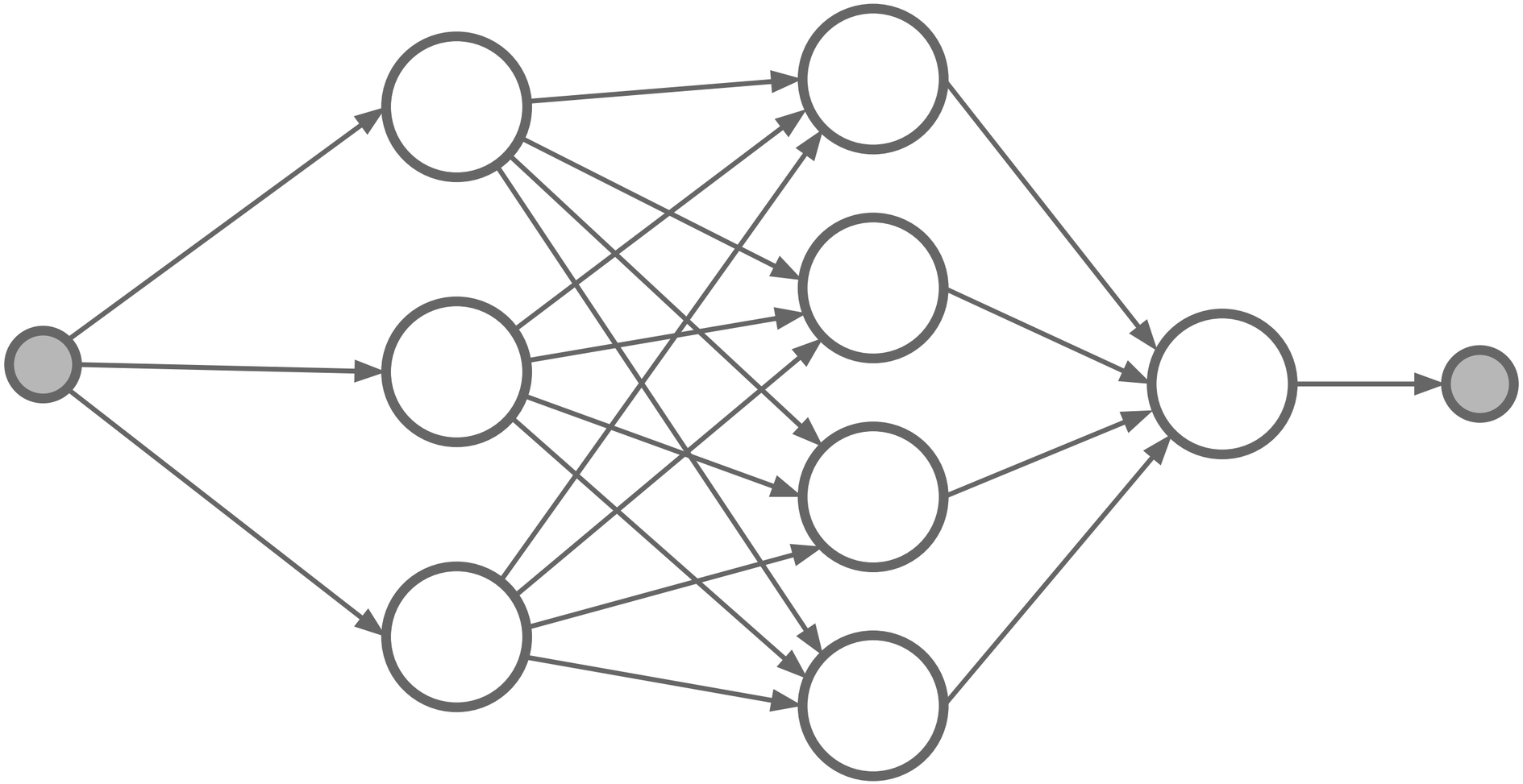
Regularização L2
(para os pesos)

$$L(\hat{y}, y) + \lambda \sum_{i=1}^{p} w_i^2$$

A regularização L2 só muda o tipo da soma que queremos minimizar, ou seja, o tipo da nossa penalização

mentorama.

Dropout *(Hinton, Srivastava, Krizhevsky, Sutskever e Salakhutdinov - 2012)*



Dropout *(Hinton, Srivastava, Krizhevsky, Sutskever e Salakhutdinov - 2012)*