Autoencoders

Eduardo de Medeiros da Silveira

Universidade Federal de Santa Maria

Representação Eficiente de Dados

Em 1970, William Chase and Herbert Simon fizeram um experimento com jogadores profissionais de xadrez, para estudar a relação entre memória, percepção e reconhecimento de padrões.

- Capazes de memorizar o tabuleiro em poucos segundos.
- Somente quando as peças estavam em posições naturais.
- O reconhecimento de padrões ajuda na memorização.

Representação Eficiente de Dados

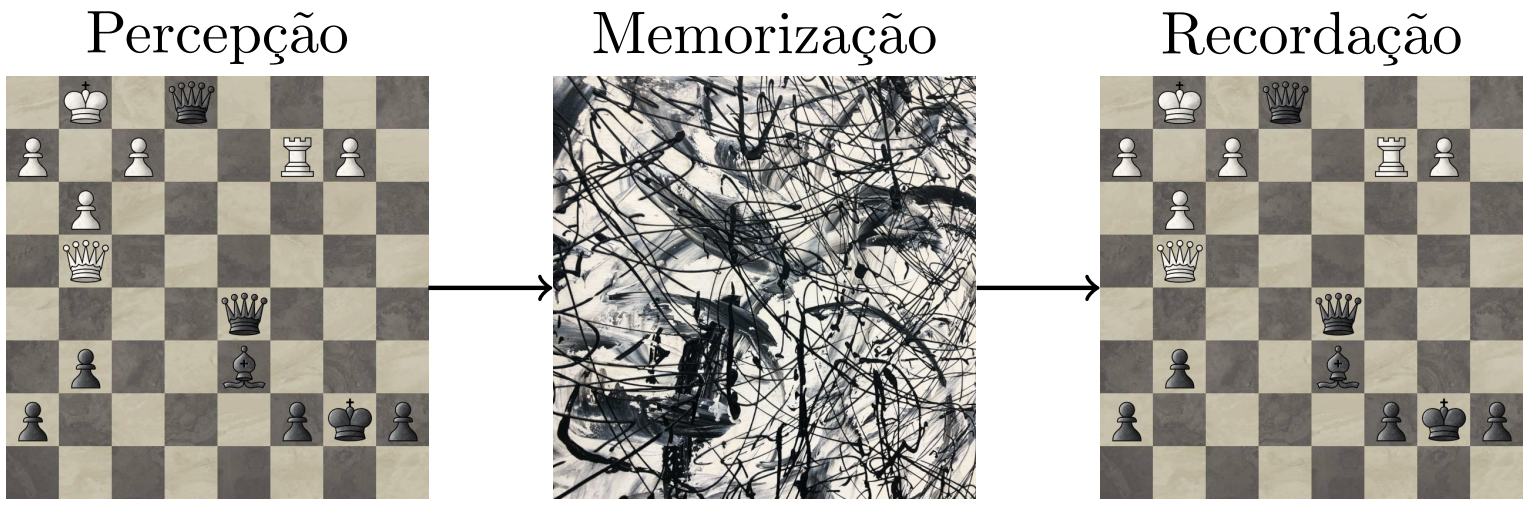

Figura: Etapas do experimento da memória no xadrez.

Manifold Hypothesis

- Conjuntos de dados com várias dimensões existem em *manifolds* de menor dimensão.

Autoencoder

Um *autoencoder* é uma rede neural que **tenta** aprender a função identidade.

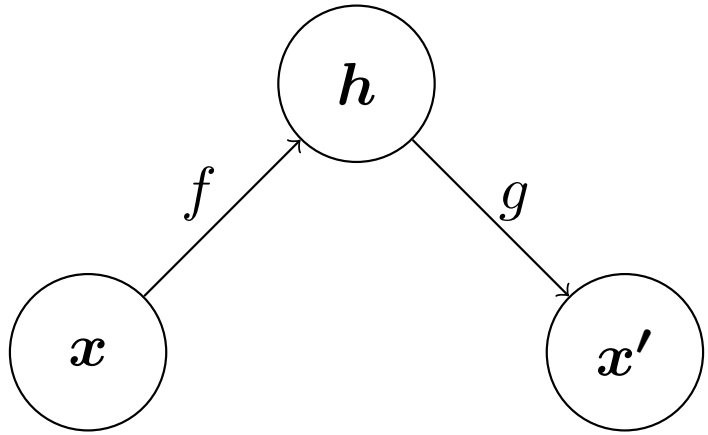

Figura: Esquema geral de um *autoencoder*, que mapeia uma entrada $\boldsymbol{x}$ para uma saída $\boldsymbol{x'}$, através de uma representação interna $\boldsymbol{h}$. O *autoencoder* é composto por um codificador f e um decodificador g.

Autoencoder

Algumas características:

- Aprendizado não-supervisionado ou auto-supervisionado.
- A saída não importa.
- A representação latente $\boldsymbol{h}$ importa.
- Restrições.

Autoencoder

Algumas características:

- Mesmo número de neurônios na entrada e na saída.
- Geralmente é simétrico.
- *Stacked* ou *deep autoencoders*.
- Nem sempre $\boldsymbol{h}$ vai capturar informações importantes.

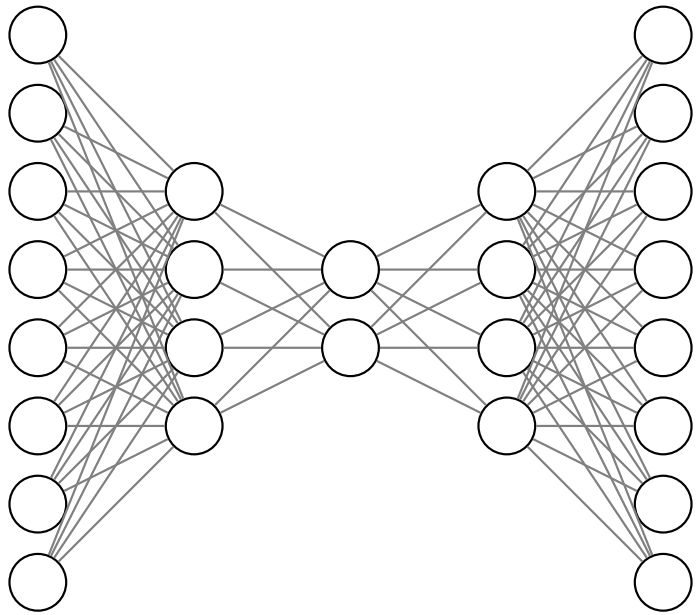

Figura: Exemplo de um autoencoder.

Undercomplete Autoencoder

- A dimensão de $\boldsymbol{h}$ é menor do que a dimensão de $\boldsymbol{x}$.
- Minimiza-se L, que calcula a dissimilaridade de $\boldsymbol{x}$ e $\boldsymbol{x'}$.

$$L(\boldsymbol{x}, \boldsymbol{x'}) = L(\boldsymbol{x}, g(f(\boldsymbol{x})))$$

Pré-treino não-supervisionado

- Queremos treinar um modelo supervisionado.
- Temos poucas observações rotuladas.
- Podemos treinar um *autoencoder* e reutilizar o *encoder*.
- Congelamento dos parâmetros.

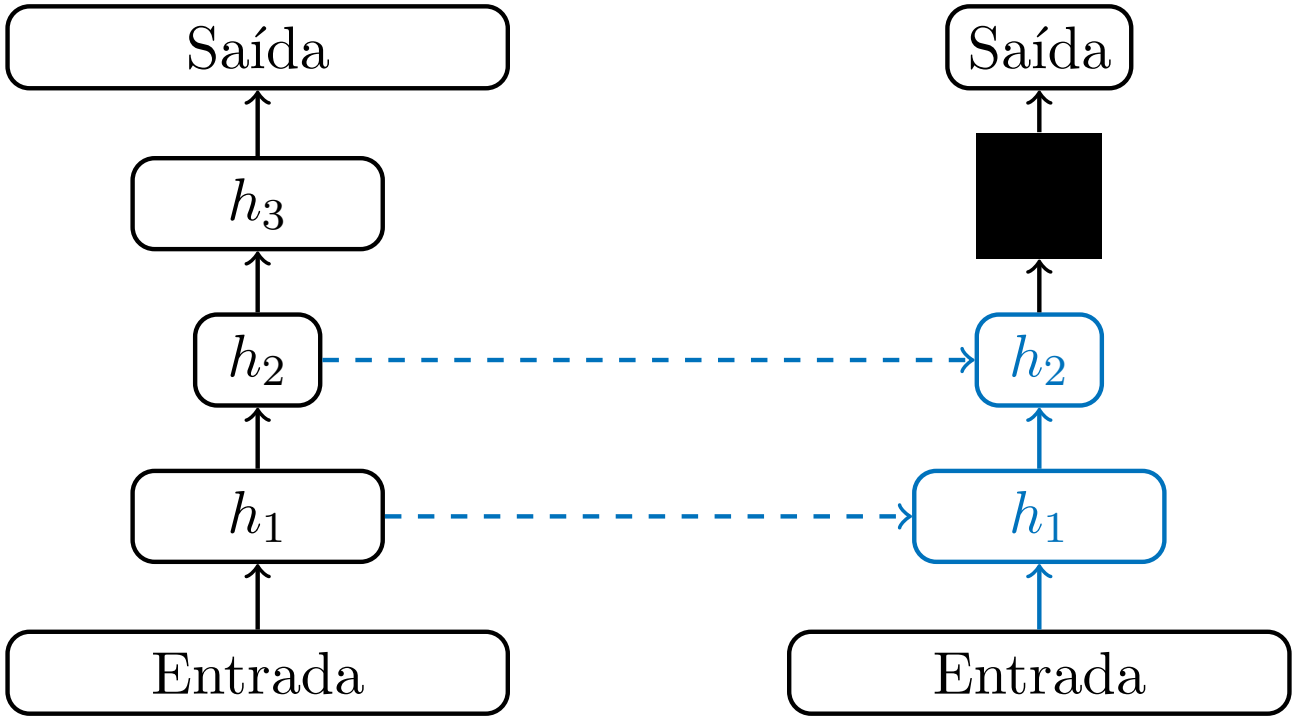

Figura: Aproveitamento da função de *encoding* em outra rede neural.

Enlace de parâmetros

- Se o *autoencoder* for simétrico, podemos criar um enlace entre os parâmetros do *encoder* e *decoder*.
- Reduzimos pela metade o número de parâmetros.
- O vetor de viés é mantido.

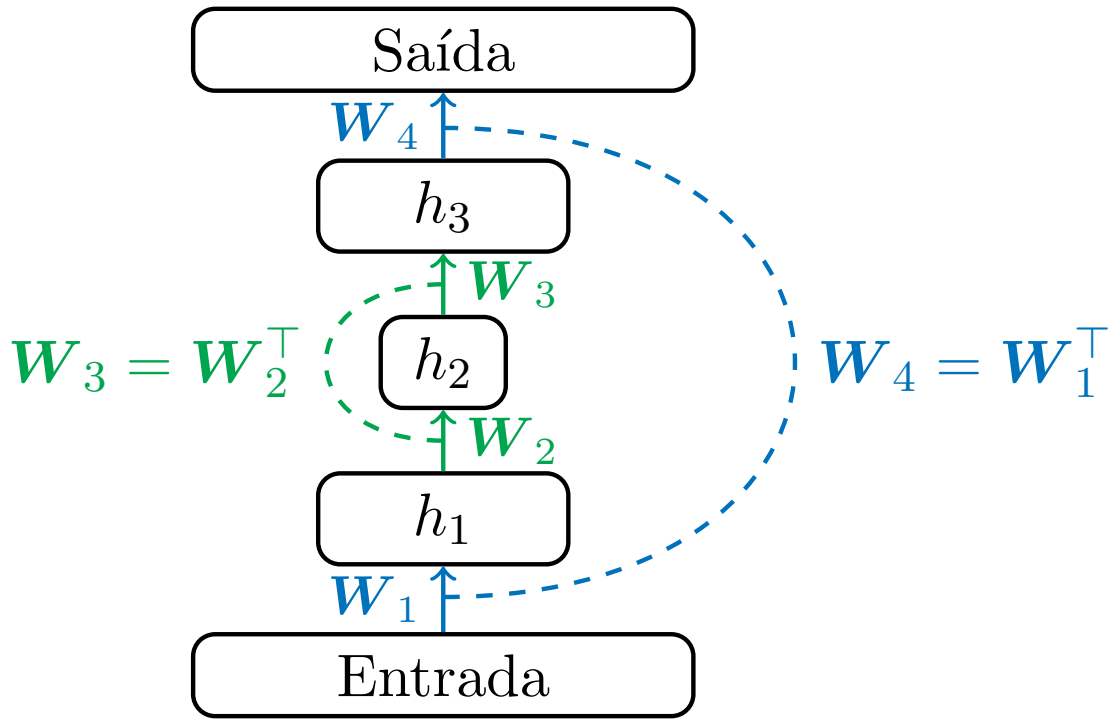

Figura: Enlace dos parâmetros de um *encoder* e um *decoder*.

Treinamento por Camadas

Em vez de treinar todo o *autoencoder*, podemos dividir o treino por camadas.

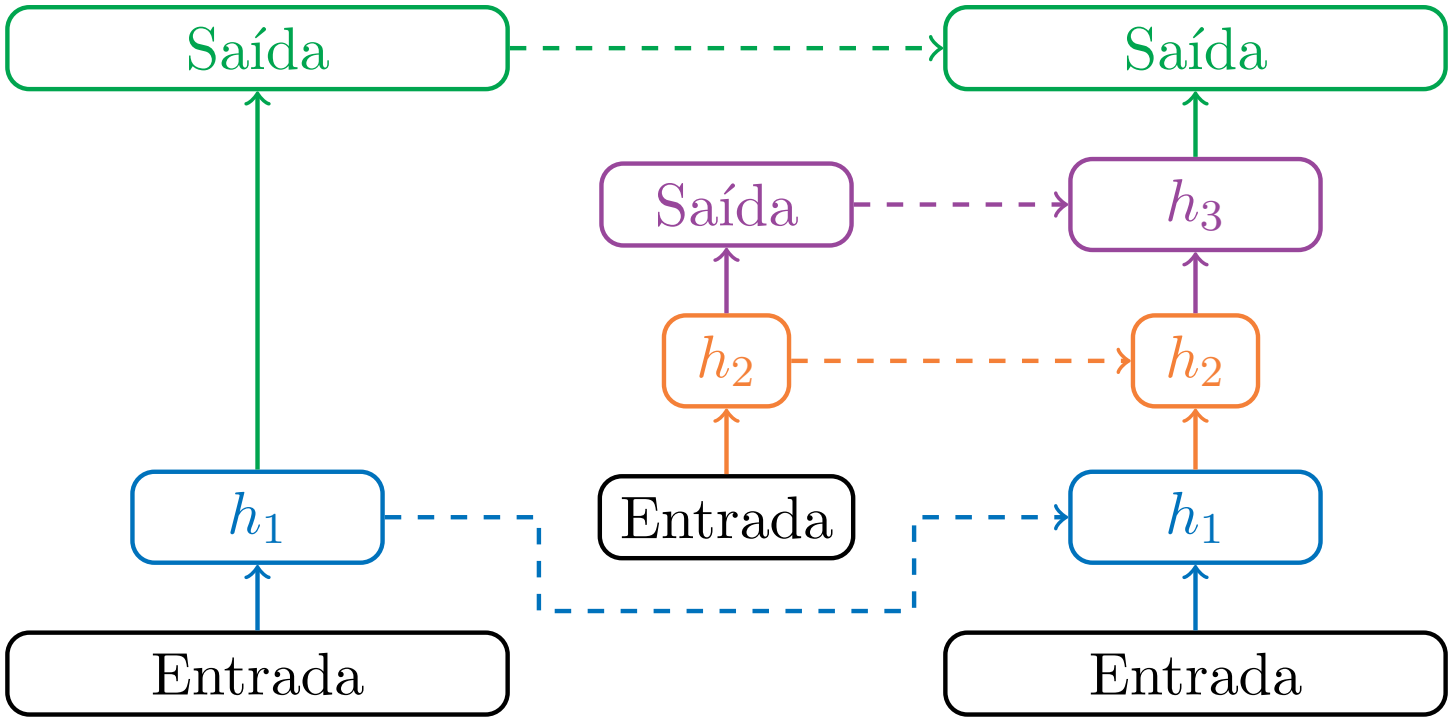

Figura: Treinamento por camadas de um *autoencoder*. Primeiramente, treinamos as camadas h_1 e Saída. Depois, treinamos h_2 e h_3.

Convolutional Autoencoders

- Camadas convolucionais em vez de camadas densas.
- Principalmente para imagens.
- O *encoder* faz a compressão da imagem, enquanto o *decoder* faz a decompressão (camadas transpostas).

Overcomplete Autoencoders

- A representação latente $\boldsymbol{h}$ tem dimensão maior ou igual à da entrada.

Denoising Autoencoders

- Adição de ruído na entrada (Gaussiano ou Dropout).
- Treinamento para recuperar os dados originais.

$$L(\boldsymbol{x}, g(f(\tilde{\boldsymbol{x}})))$$

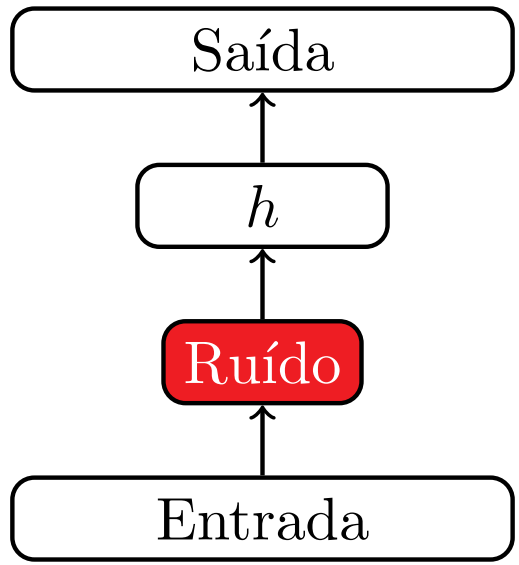

Figura: Esquema de um *denoising autoencoder*. O ruído pode representar a adição de ruído Gaussiano aos dados ou uma camada de Dropout.

Sparse Autoencoders

- A representação latente é penalizada por sua esparsidade.
- Escolhemos uma quantidade de neurônios que queremos ativados.
- Cada neurônio acaba representando uma característica específica.

$$L(\boldsymbol{x}, \boldsymbol{x'}) + \Omega(\boldsymbol{h}) = L(\boldsymbol{x}, g(f(\boldsymbol{x}))) + \Omega(\boldsymbol{h})$$