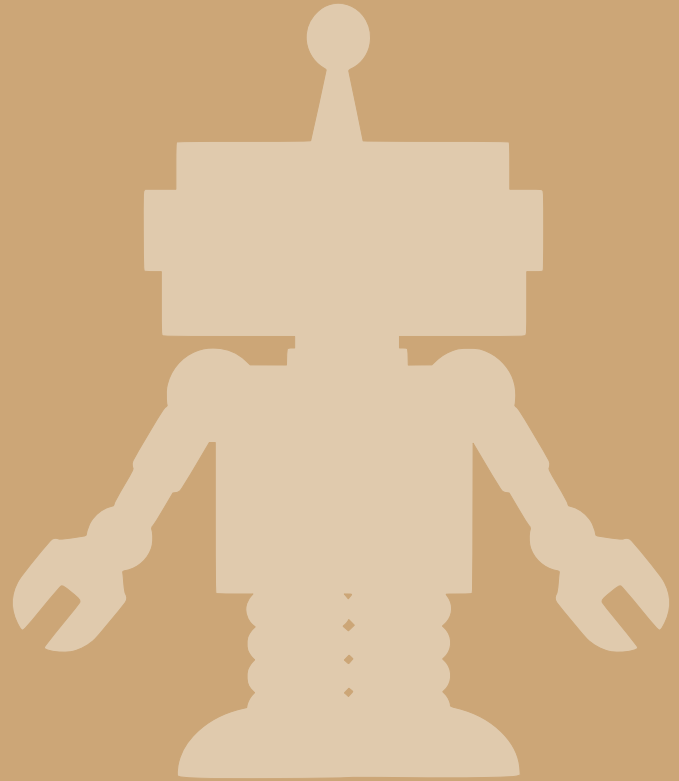


Aprendizado de Máquina 2

Aula 7

Professora: Patrícia Pampanelli

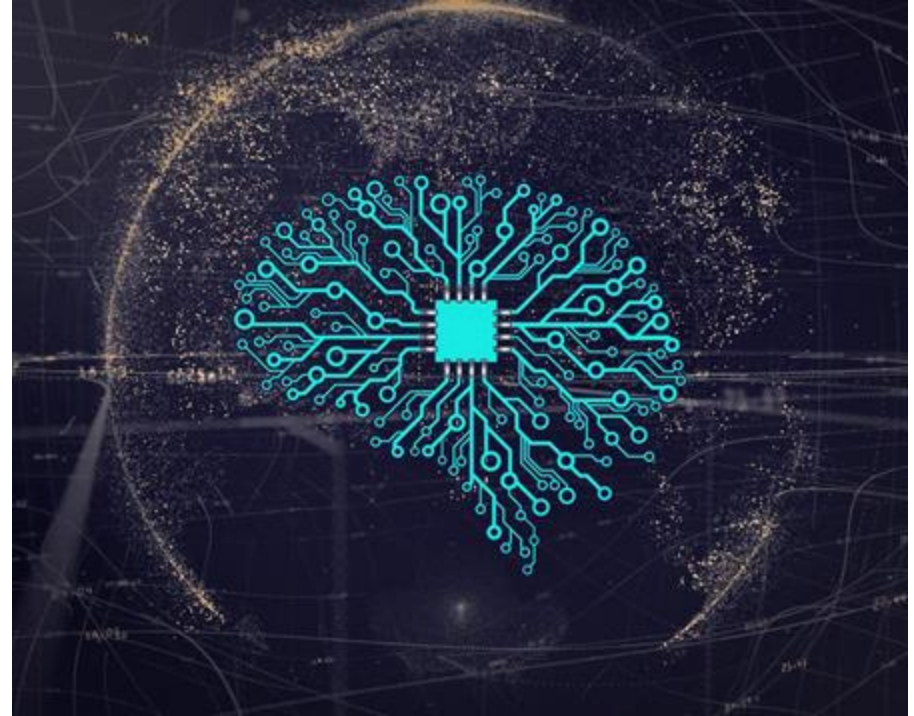
patricia.pampanelli@usp.br



Dúvidas da última aula?

Aula de Hoje

- Auto encoders



*Redução de dimensionalidade
consistem comprimir um conjunto
grande de features em um subespaço
de dimensão menor sem que haja
perda de informação*



Redução de Dimensionalidade

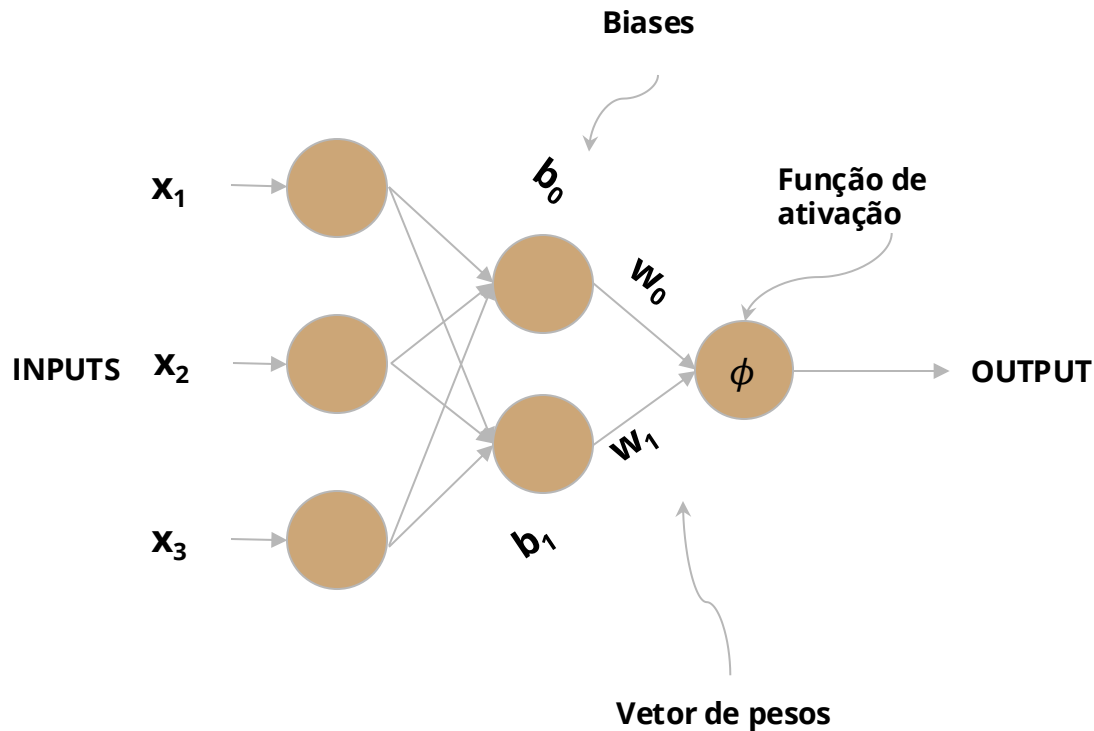
- Vocês já tiveram contato com algumas abordagens para redução de dimensionalidade:
 - K-means
 - Principal component analysis
 - Self-organizing maps
- Hoje nós vamos falar sobre outra técnica para redução de dimensionalidade: **autoencoders**
- Esta é uma técnica baseada em redes neurais artificiais



Recapitulando... Redes neurais

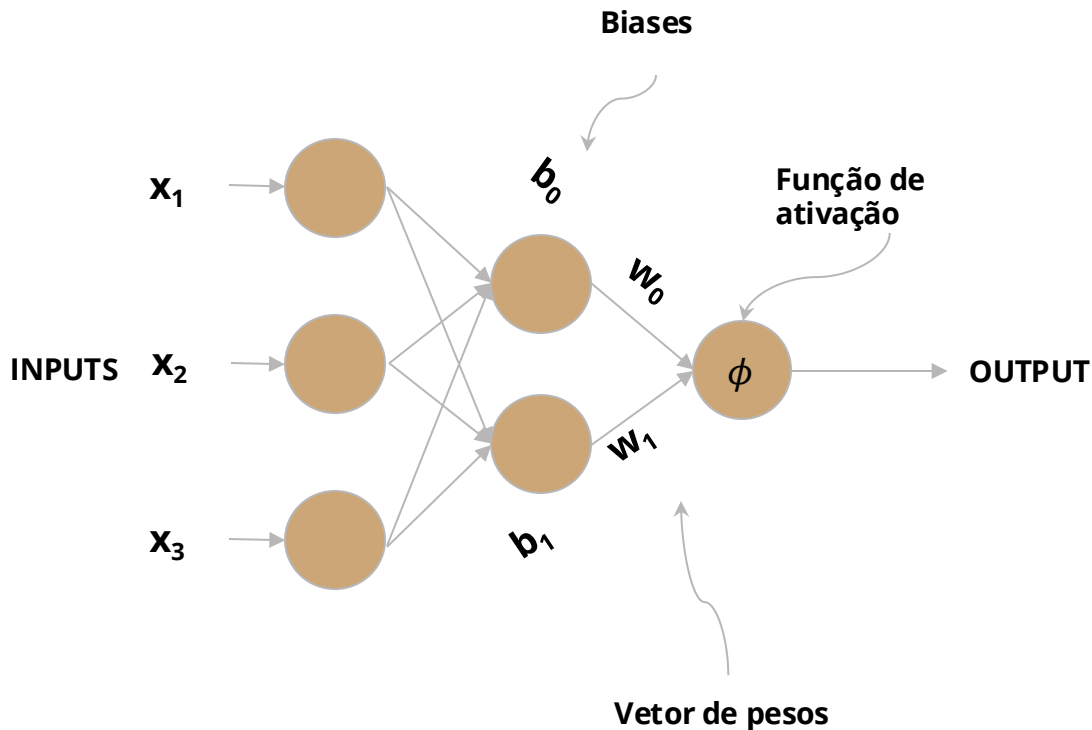
Redes neurais

Esta é uma visão esquemática de uma camada com 3 neurônios e aplicação de funções de ativação.



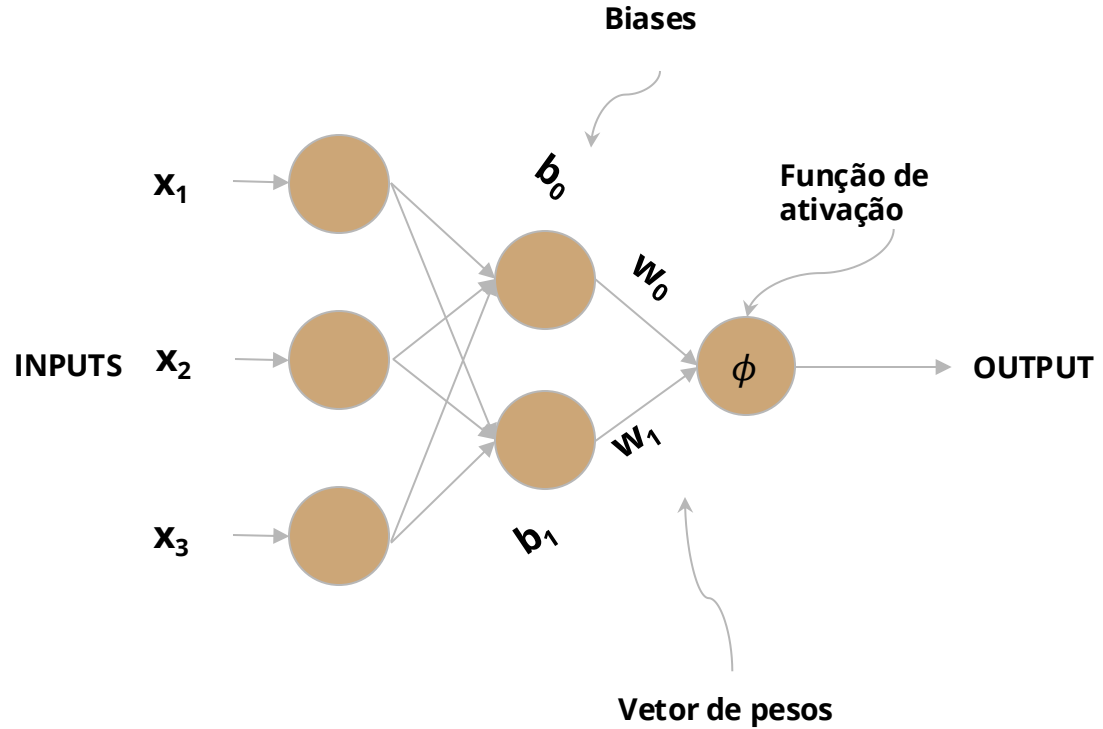
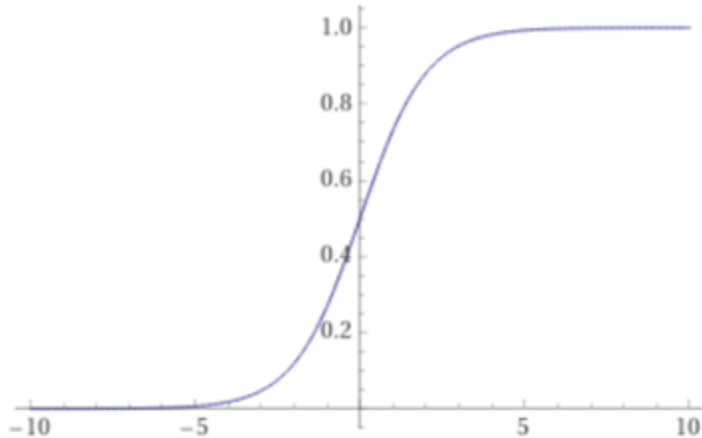
Redes neurais

- Funções de ativação:
 - Em 1986, David Rumelhart, Geoffrey Hinton, e Ronald Williams publicaram um trabalho propondo o uso do *backpropagation* como algoritmo de treinamento das redes neurais.
 - Para utilização deste algoritmo de treinamento, a derivada da função de ativação é importante. Desta forma, a *step function* foi substituída por outras funções de ativação com derivadas mais fáceis.



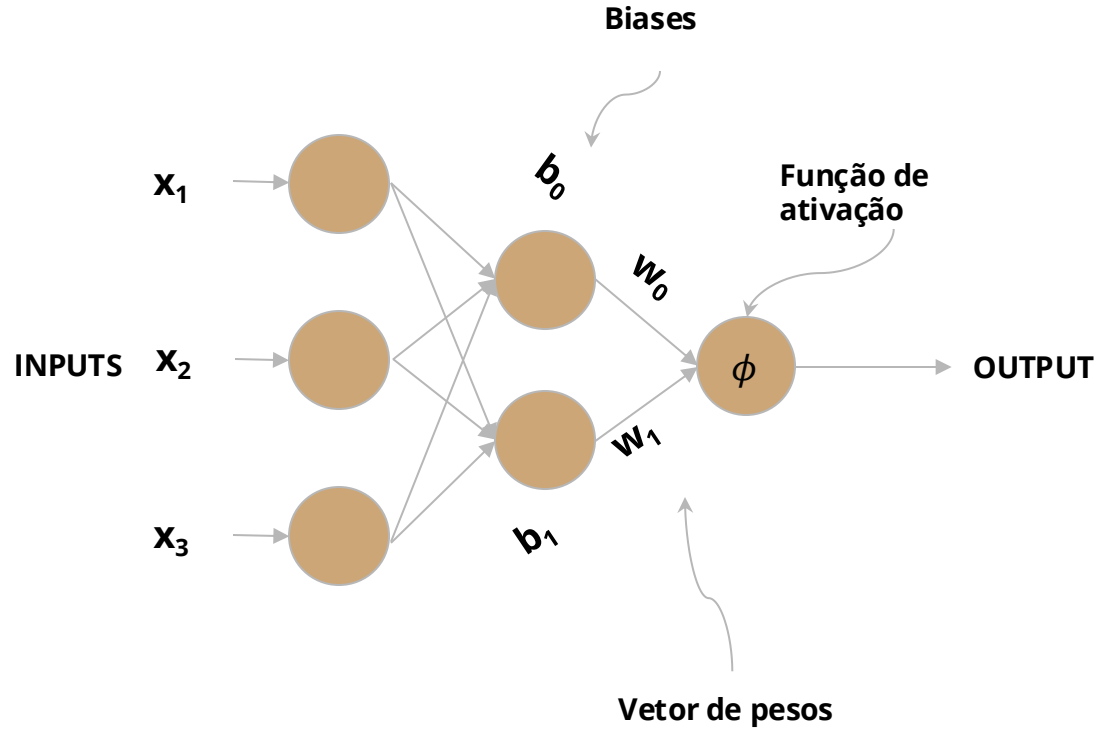
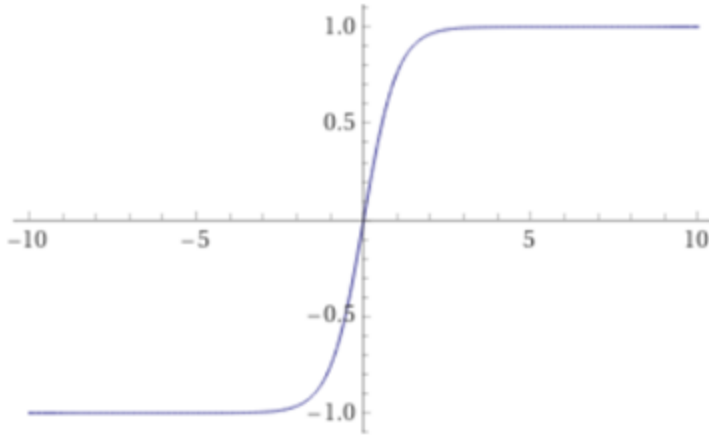
Redes neurais

- Funções de ativação:
 - Sigmoide



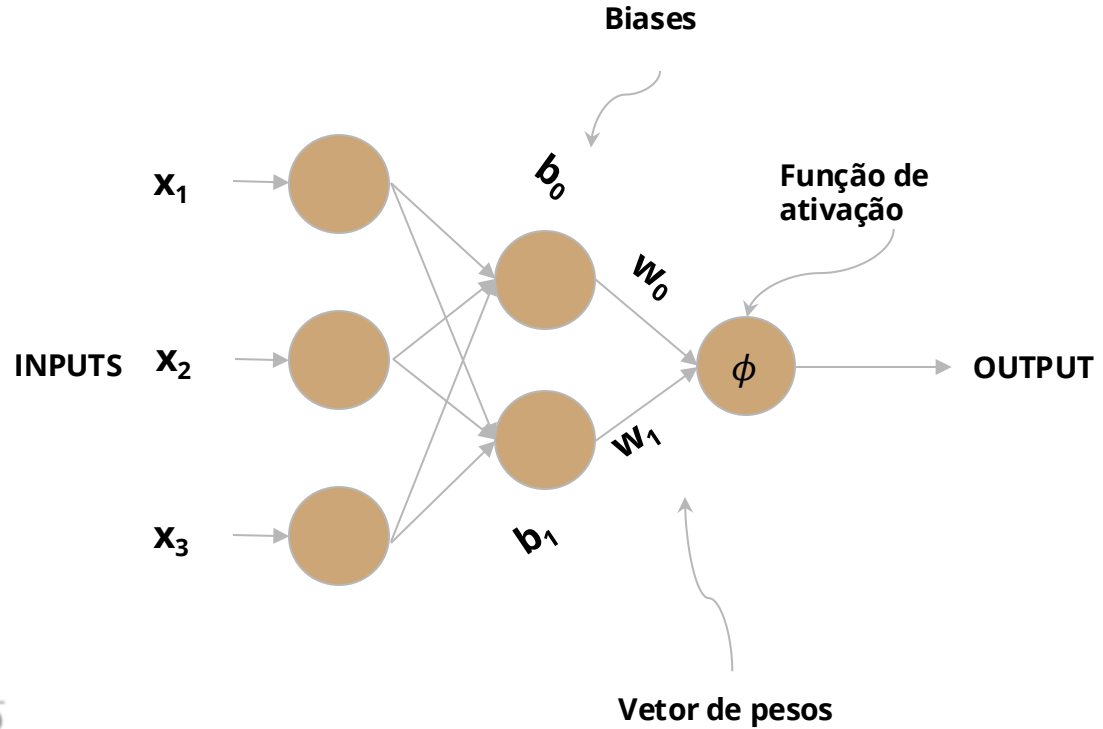
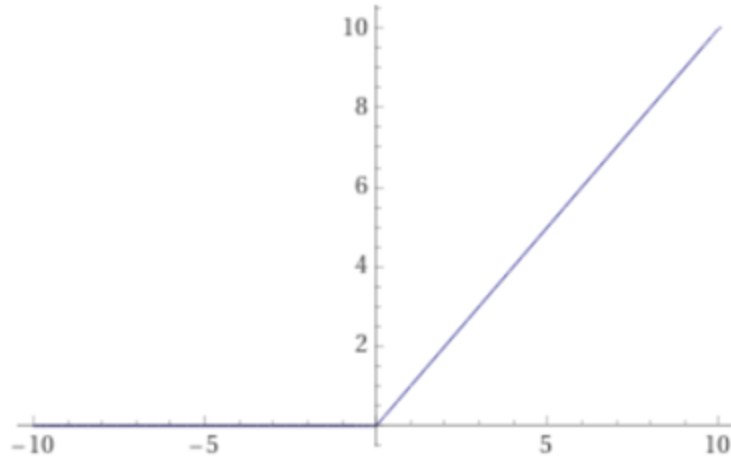
Redes neurais

- Funções de ativação:
 - Tangente hiperbolica



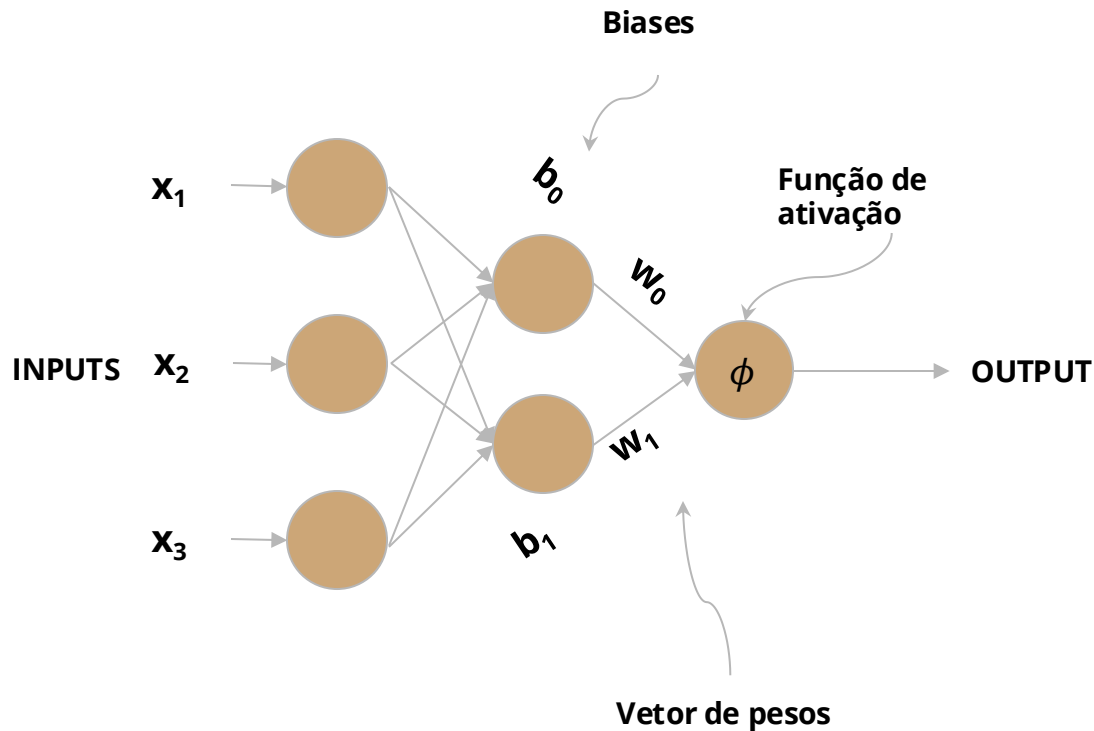
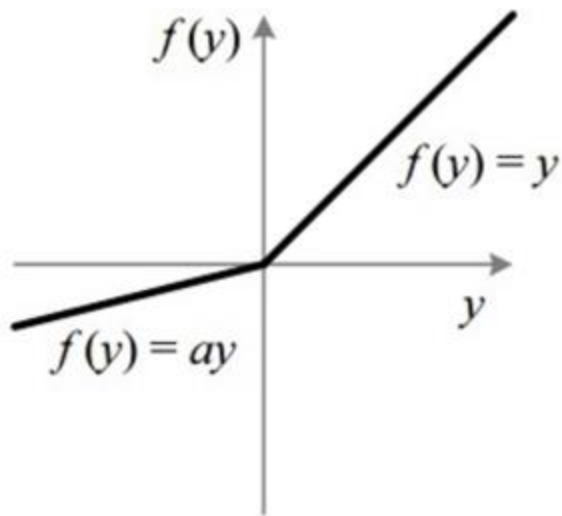
Redes neurais

- Funções de ativação:
 - ReLU



Redes neurais

- Funções de ativação:
 - Leaky ReLU



Redes neurais

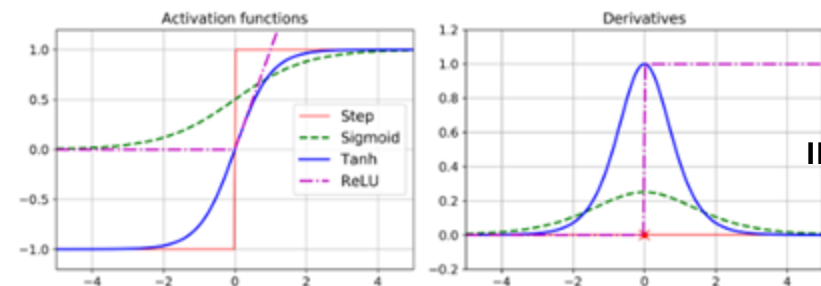
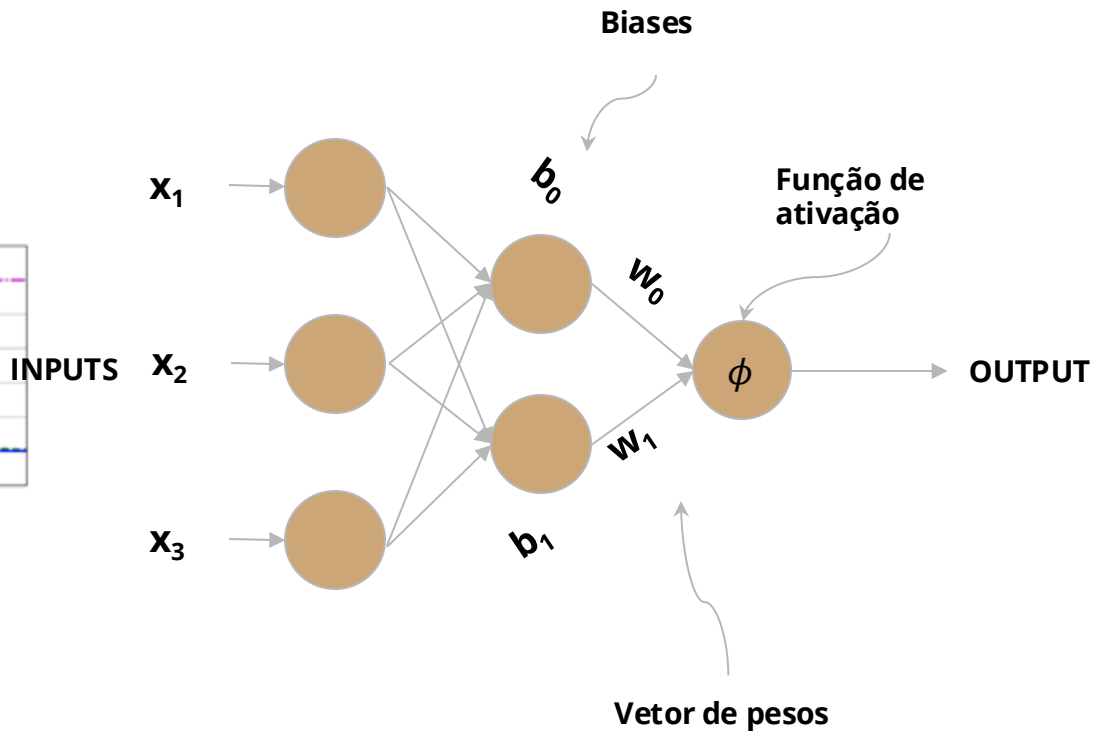


Figure 10-8. Activation functions and their derivatives



Identificação de padrões

Representação eficiente de padrões

- A relação entre memória, percepção e identificação de padrões foi estudada pela primeira vez nos anos de 1970s por William Chase e Hebert Simon
- Eles observaram que jogadores profissionais de xadrez podiam memorizar a configuração das peças do tabuleiro observando-o por somente 5 segundos
- Eles só eram capazes de memorizar caso as peças estivessem em uma configuração possível



Representação eficiente de padrões

- Os jogadores de xadrez não tem uma memória diferente de qualquer outro ser humano.
- A diferença é que eles aprenderam a identificar estes padrões.



Conceitos gerais

Conceitos gerais

- Autoencoders são um tipo de **rede neural artificial** capaz de aprender padrões a partir dos dados de entrada
- Os padrões são chamados de codificação ou representação latente
- O treinamento destas redes é feito de forma **não supervisionada**



Conceitos gerais

- O fato do treinamento ser **não supervisionado** é bastante interessante, visto que não precisamos de dados anotados
- A disponibilidade de dados costuma ser um gargalo bastante relevante em projetos de inteligência artificial. Este recurso pode ser utilizado, inclusive, em treinamentos para classificação onde a disponibilidade de dados é um gargalo. Veremos na seção de aplicações.



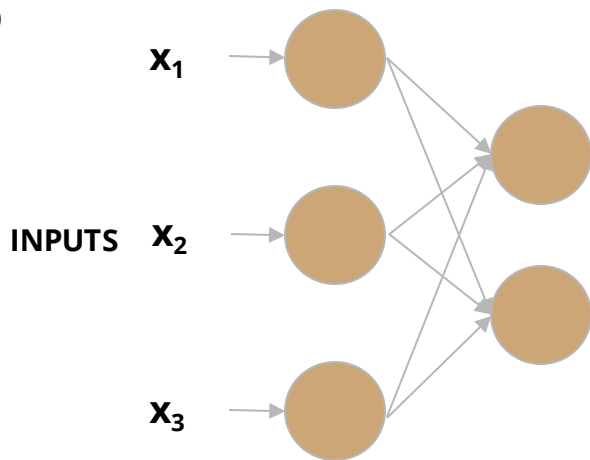
Conceitos gerais

- Autoencoders são compostos de duas partes:
 - *Encoder* (rede de reconhecimento)
 - *Decoder* (rede generativa)
- O *encoder* é responsável por converter os dados de entrada em uma representação simplificada
- O *decoder* é responsável por converter as representações simplificadas (compactadas) em outputs
- O número de neurônios da camada de entrada deve ser igual ao número de neurônios na camada de saída



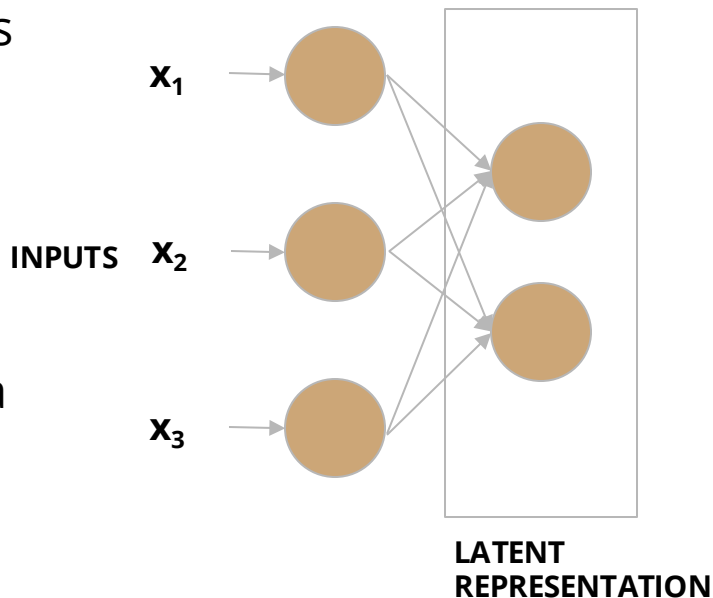
Conceitos gerais

- Os inputs para treinamento destas redes podem ser de diferentes naturezas:
 - Podemos trabalhar com imagens, dados tabulares, etc
 - Dados que queremos encontrar padrões, remover ruído ou buscar uma representação simplificada



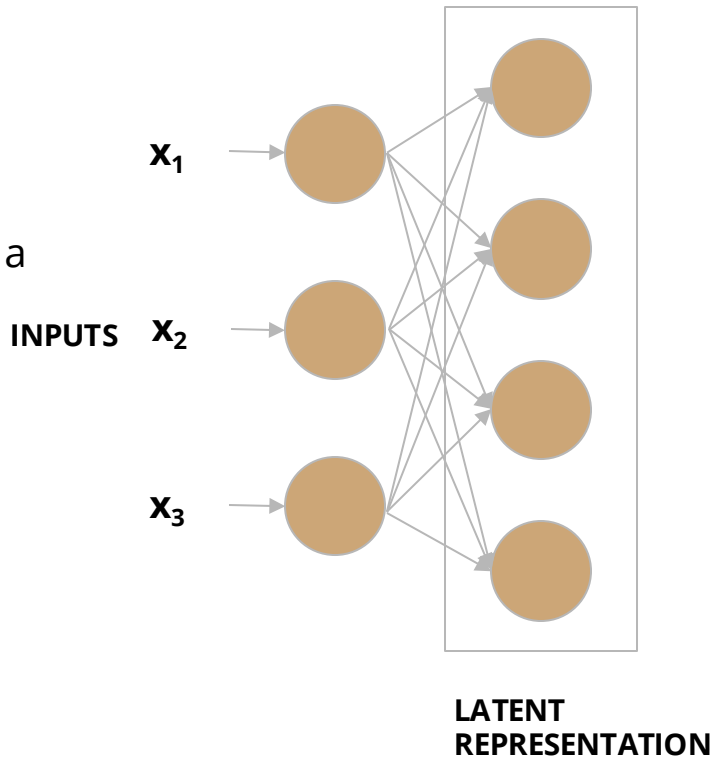
Conceitos gerais

- A representação latente dos dados consiste em uma versão reduzida dos dados de entrada.
- Na maioria das vezes esta **camada tem um número menor de neurônios** que a camada de entrada.
- Quando isso acontece o **autoencoder é chamado de incompleto**



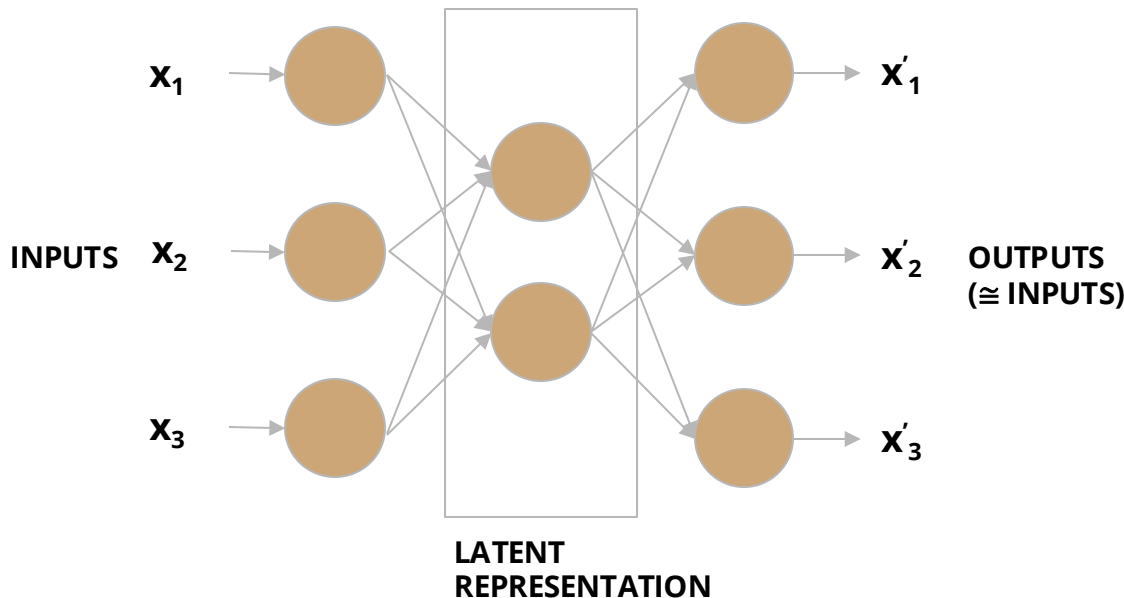
Conceitos gerais

- O autoencoder é chamado de **super completo (*overcomplete*)** quando esta camada tem um número maior de neurônios que a camada de entrada.
- Estes autoencoders são mais difíceis de serem desenhados e treinados, mas costumam ser mais poderosos ao aprender padrões mais complexos.
- Nestes casos é incluída uma camada de regularização.



Conceitos - Autoencoders

- A camada de saída tem a mesma dimensão da camada de entrada.
- O objetivo ao treinar um autoencoder é que a saída se aproxime da camada de entrada. Por isso essas redes são treinadas de forma não supervisionada.



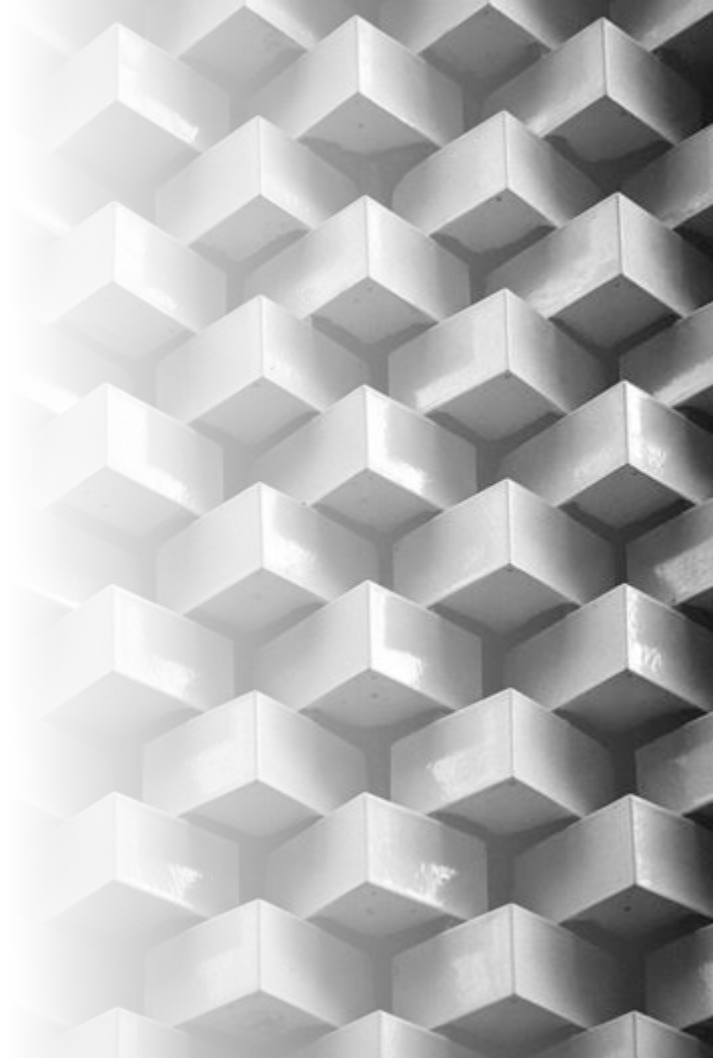
Autoencoder Simples - Iris D



Stacked Autoencoders

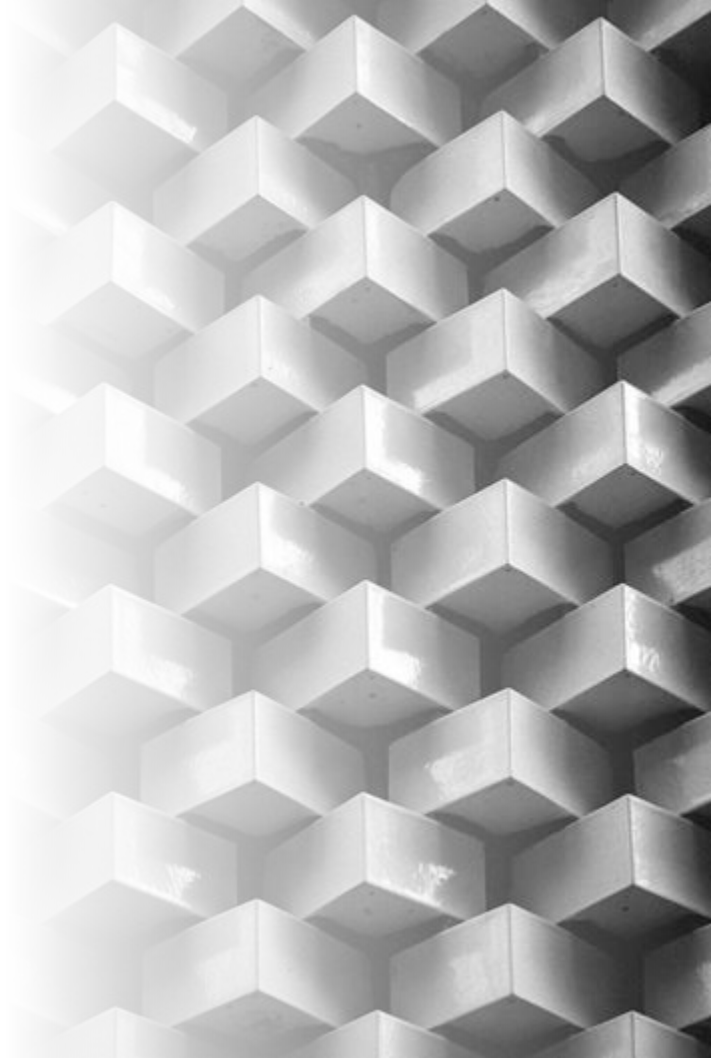
Stacked Autoencoders

- Os autoencoders podem ter **várias camadas escondidas**
- Nestes casos os autoencoders são chamados de **stacked autoencoders** (ou **deep autoencoders**)
- A adição de novas camadas ajudam a rede a aprender padrões mais complexos
- É importante ter cuidado para não fazer o autoencoders tão complexo que ele acaba mapeando cada input em um número arbitrário. Nestes casos, o autoencoder acaba não aprendendo padrões relevantes dos dados de entrada.



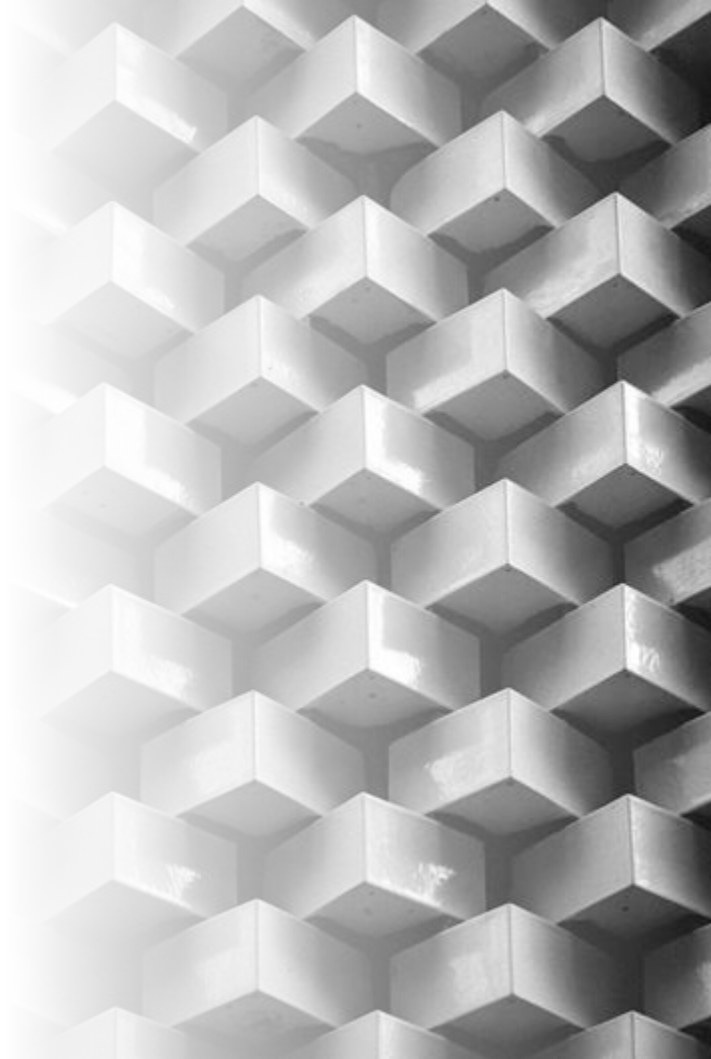
Stacked Autoencoders

- A arquitetura dos autoencoders geralmente segue uma estrutura simétrica em relação a camada central (camada de codificação).
- Simplificando, a arquitetura é como se fosse um “sanduíche”
- Vamos ver no próximo slide como ficaria uma arquitetura de um autoencoders para reduzir dimensões dos dados do MNIST

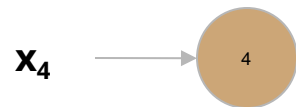
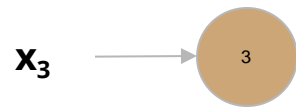
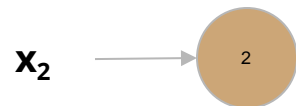
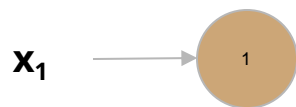


Stacked Autoencoders

- Todas as camadas do autoencoder são totalmente conectadas (*fully connected*)
- Em outras palavras, cada neurônio de uma camada está conectado a todos os neurônios da camada seguinte

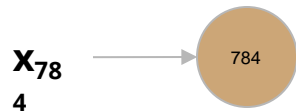


INPUTS



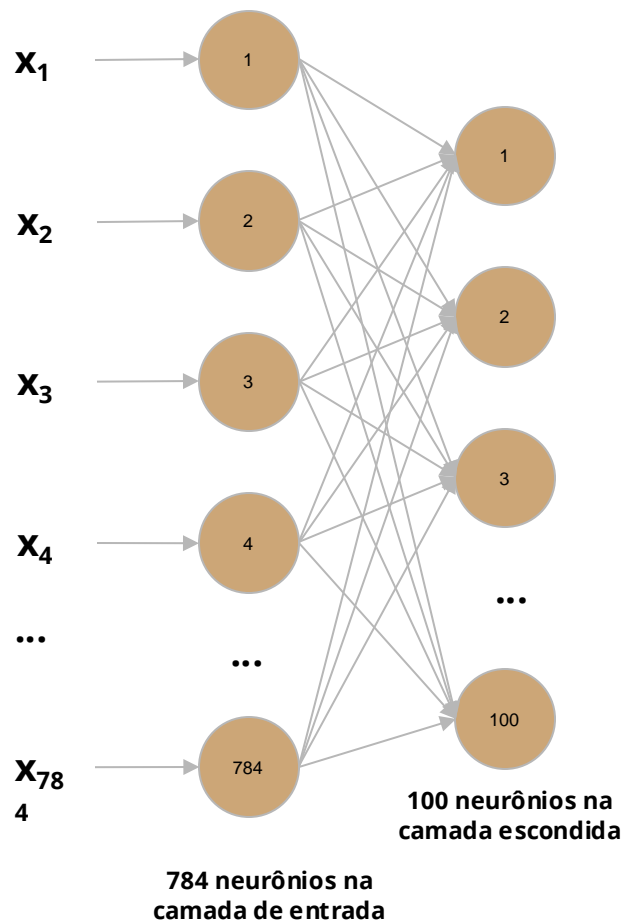
...

...

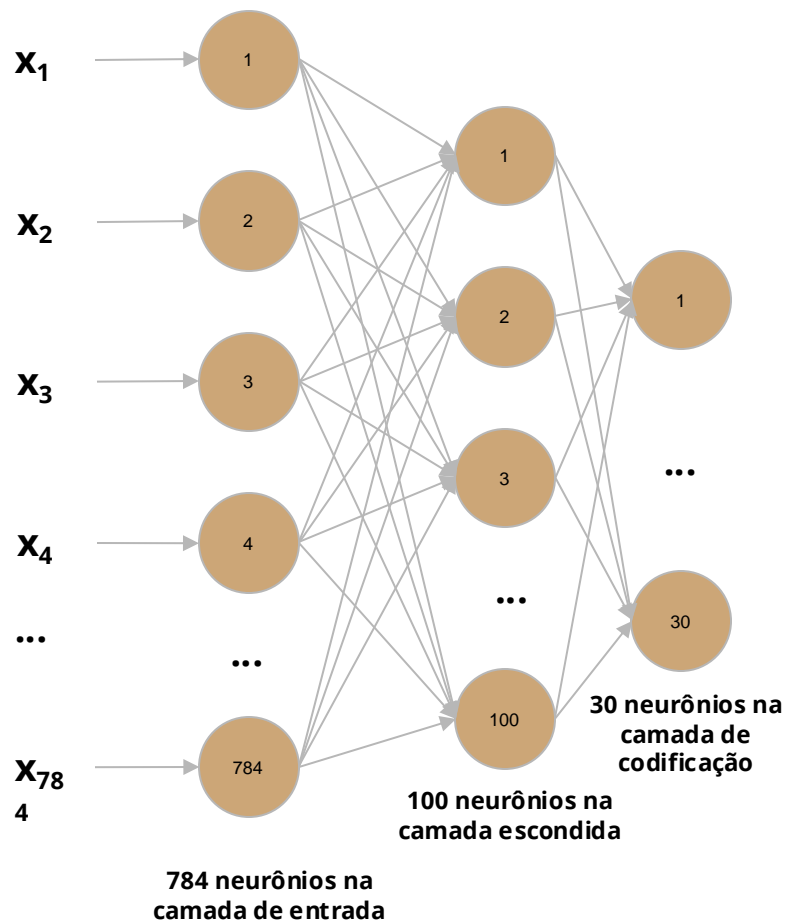


784 neurônios na
camada de entrada

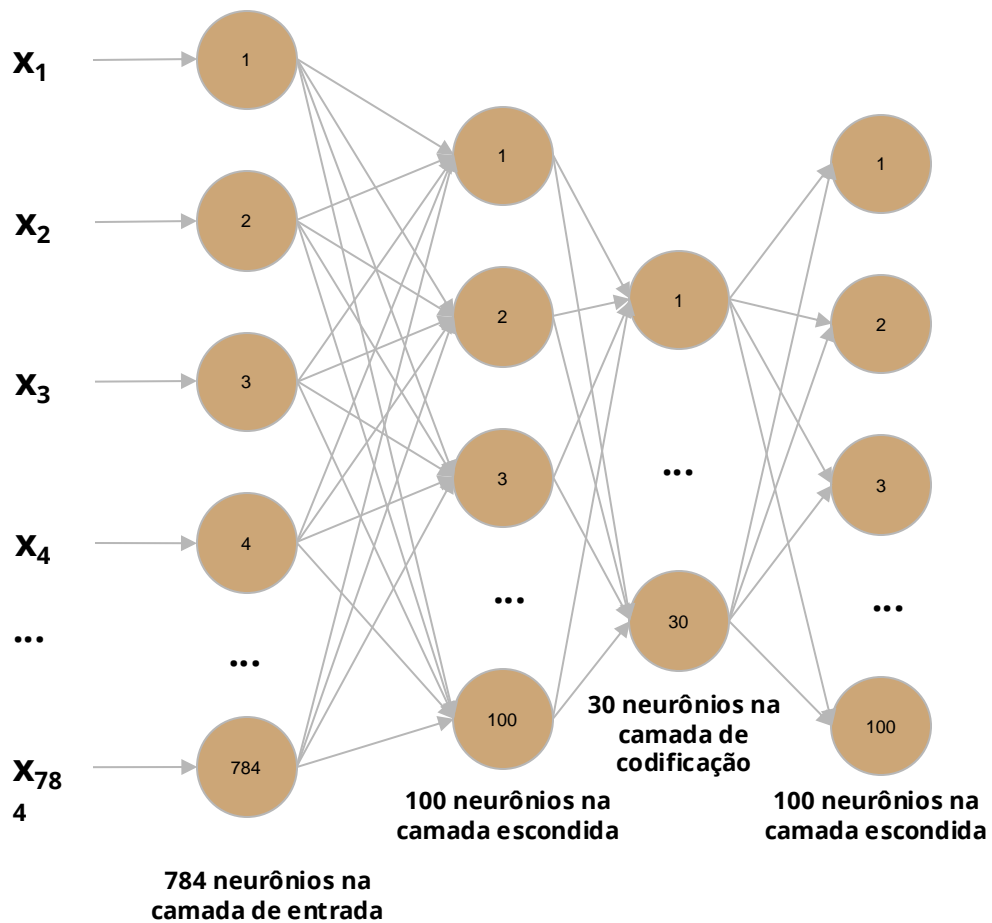
INPUTS



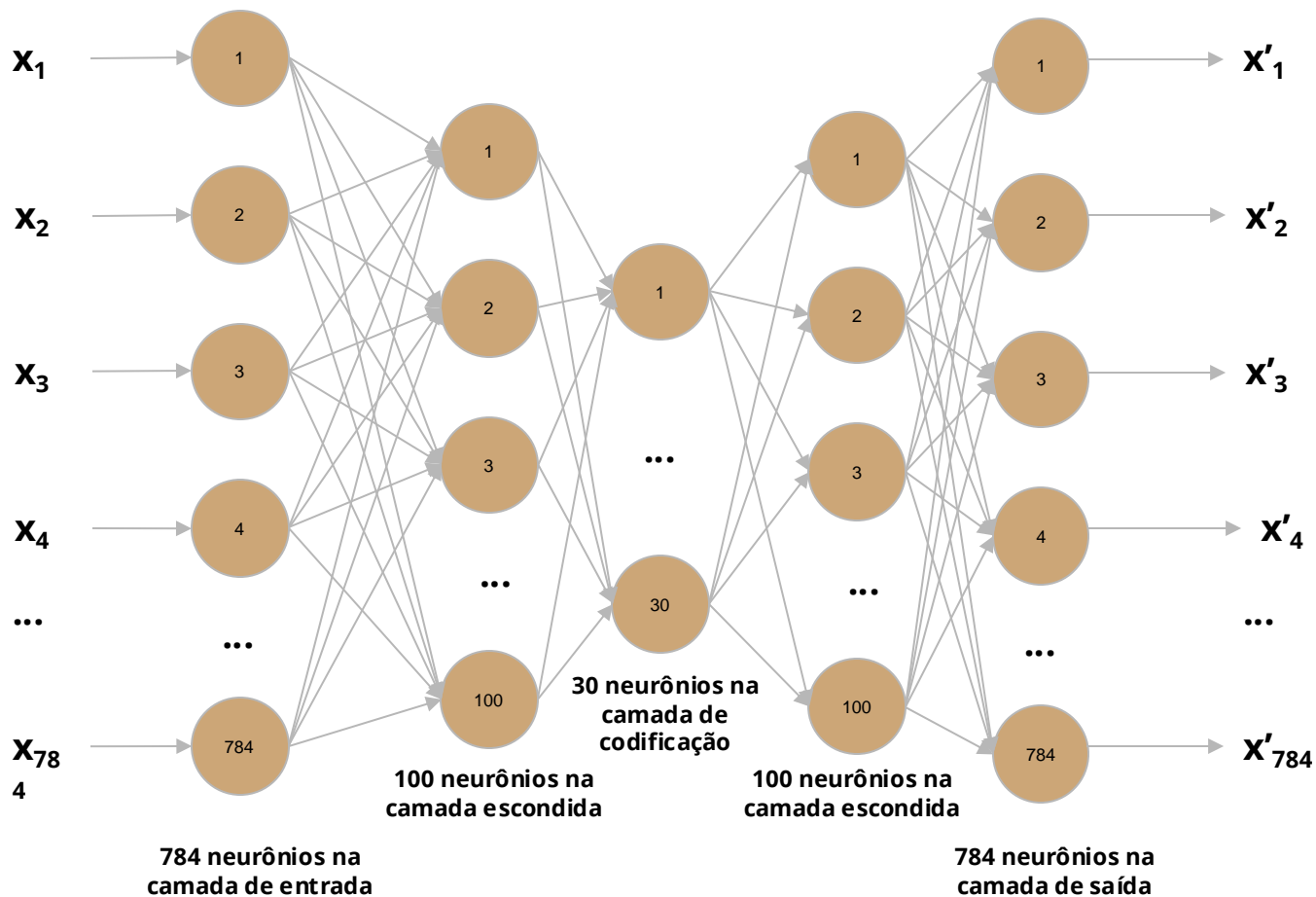
INPUTS



INPUTS



INPUTS



Autoencoders

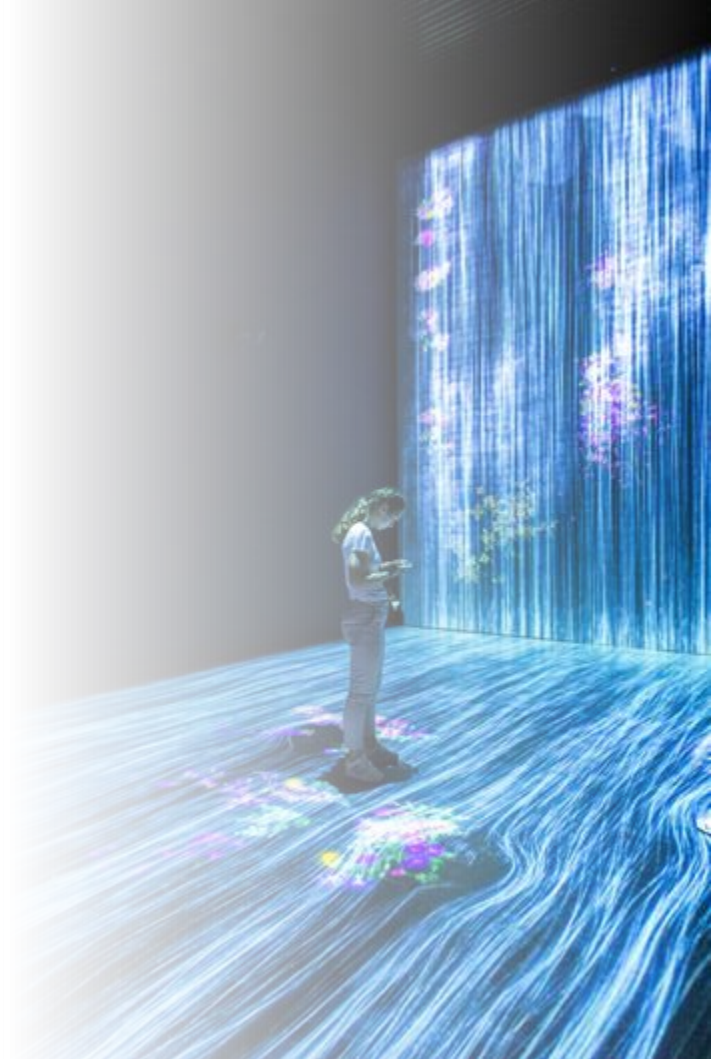
- Autoencoders são ferramentas extremamente flexíveis em Deep Learning!
- Redução de dimensionalidade é uma das aplicações destes modelos baseados em redes neurais.
- Mudanças aplicadas às saídas da rede, por exemplo, podem tornar os autoencoders ferramentas interessantes para outras finalidades.
- Nos próximos blocos de aula vamos entender algumas dessas modificações.



Pré-treinamento não
supervisionado

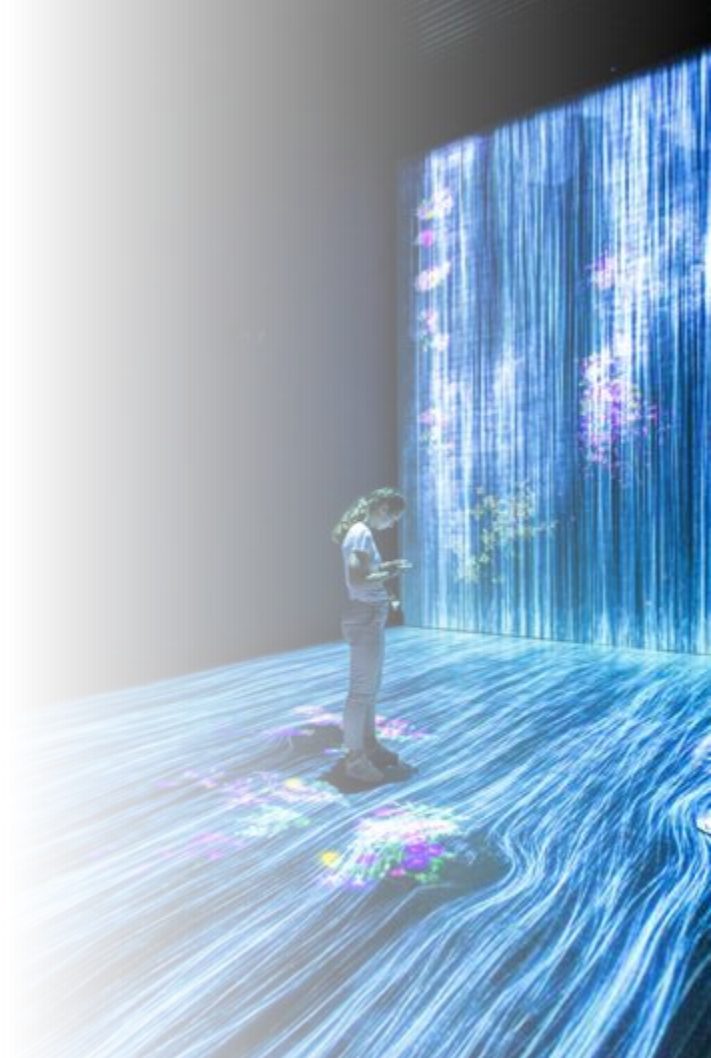
Pré-treinamento não supervisionado

- Uma utilização bastante interessante de **autoencoders** é para fazer **treinamentos de modelos de classificação** em cenários onde você **não tem muitos dados anotados**.
- Este é um cenário bastante comum, visto que é sempre mais barato obter grandes volumes de dados não anotados. A anotação de dados é uma tarefa bastante custosa.



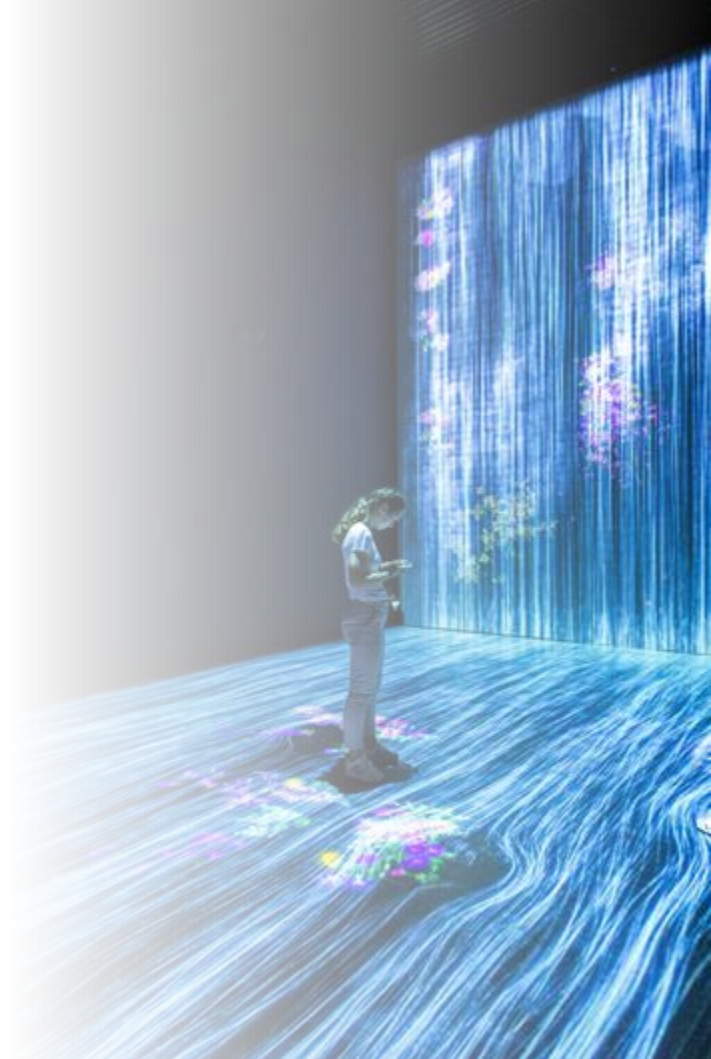
Pré-treinamento não supervisionado

- Se a maioria dos seus dados são **não anotados (*unlabeled*)**, você pode utilizar o autoencoder para aprender as **características mais importantes** dos seus dados.
- Posteriormente, pode **alterar as últimas camadas** da etapa de decoder substituí-las por **uma camada de saída e uma função softmax**.

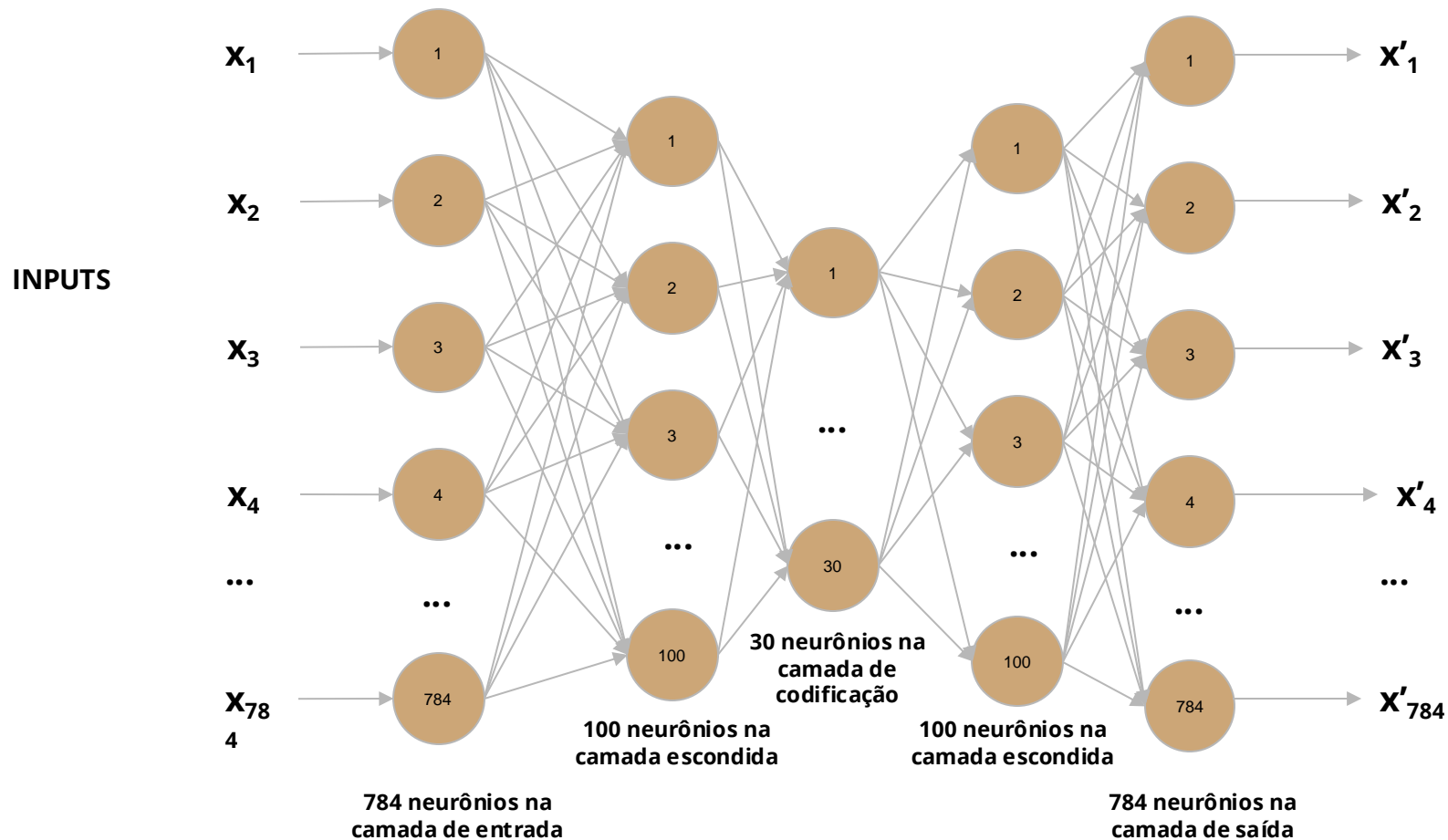


Pré-treinamento não supervisionado

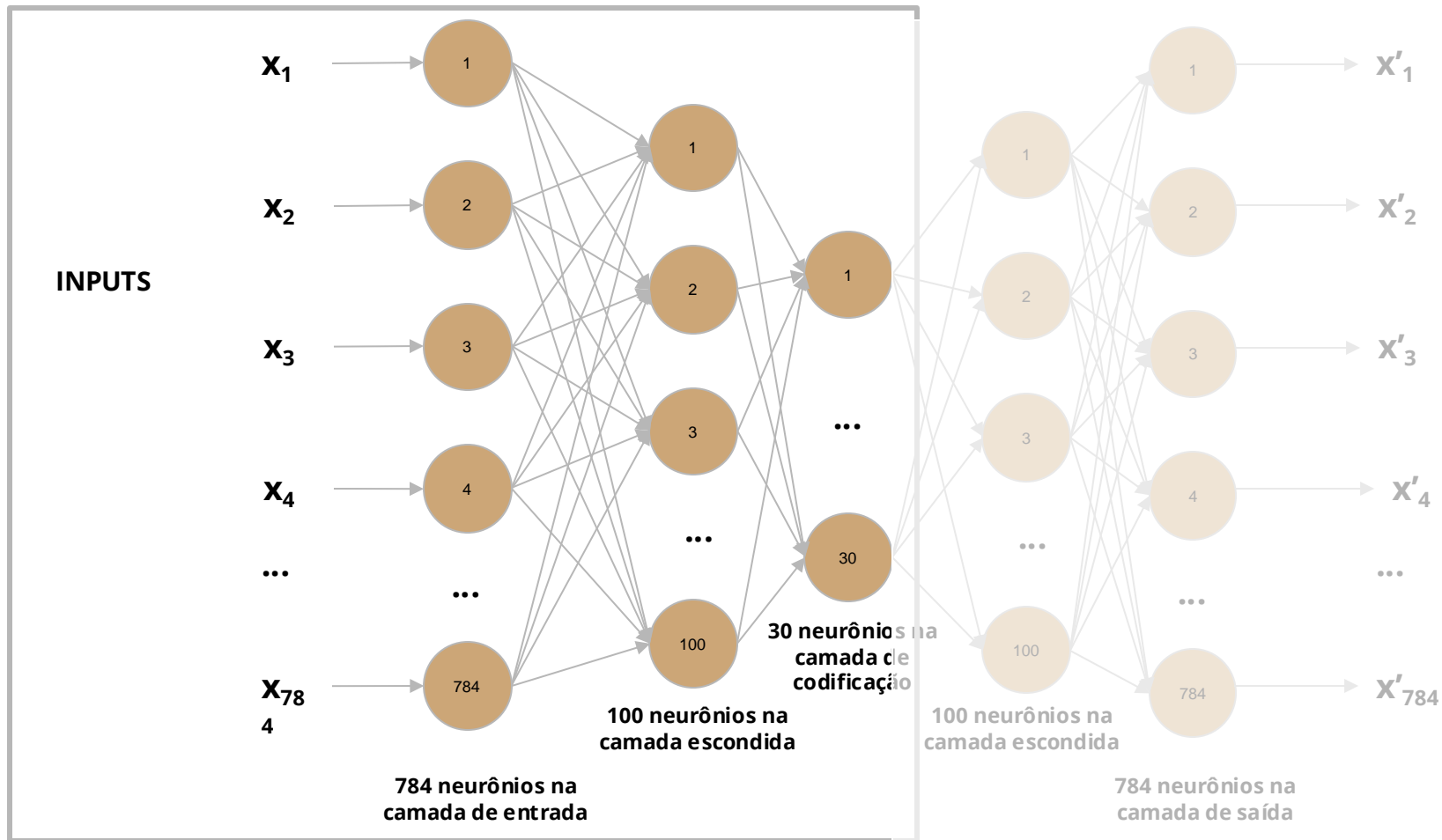
- Vamos utilizar o exemplo do MNIST mostrado anteriormente. Neste exemplo, o autoencoder é treinado com um conjunto de dados de dígitos escritos à mão.
- Apesar do dataset set anotado, os **labels** aqui **não são necessários para treinar o autoencoder**



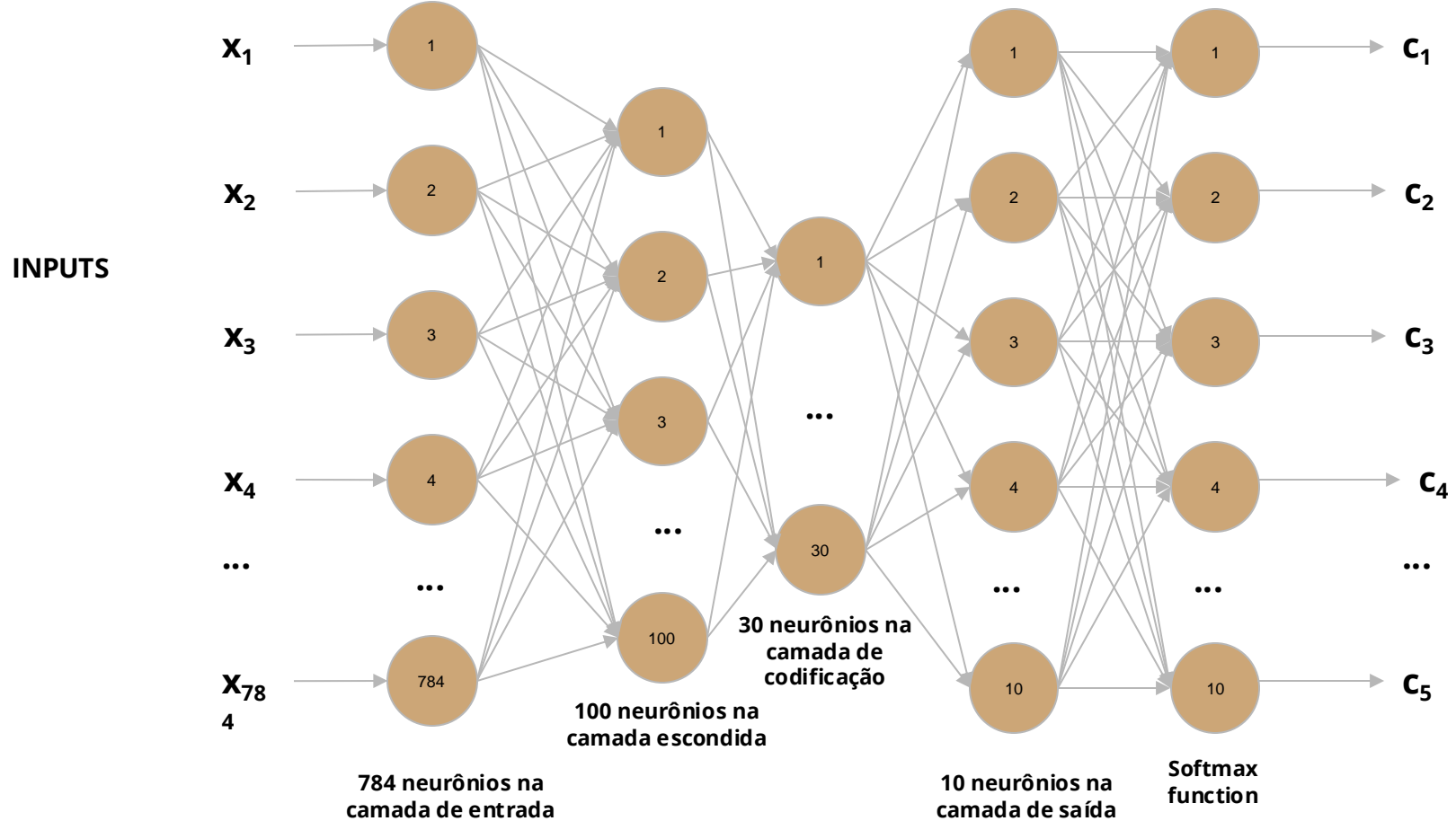
Stacked autoencoder treinado de forma não-supervisionada



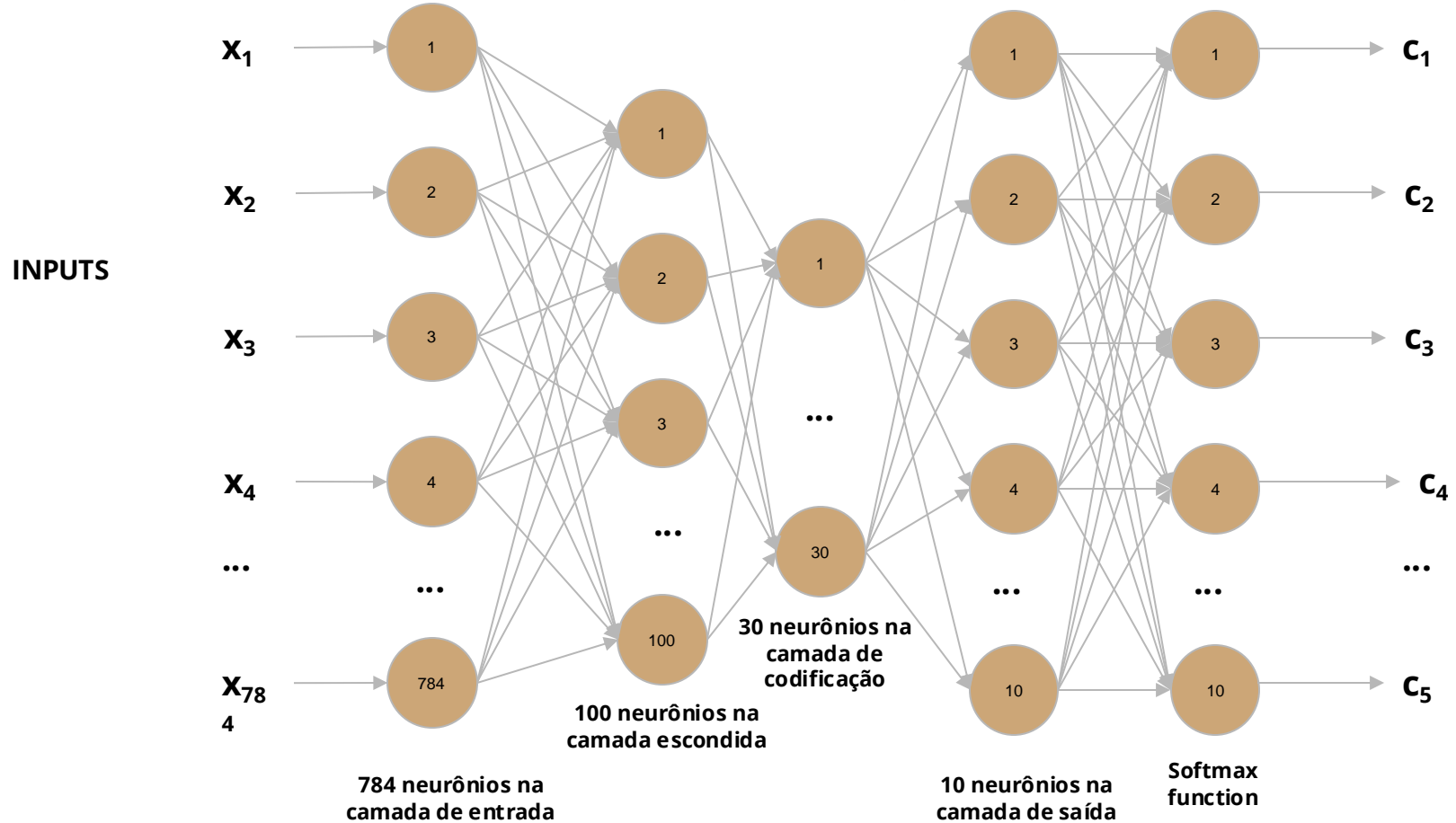
Utilizar as primeiras camadas da rede



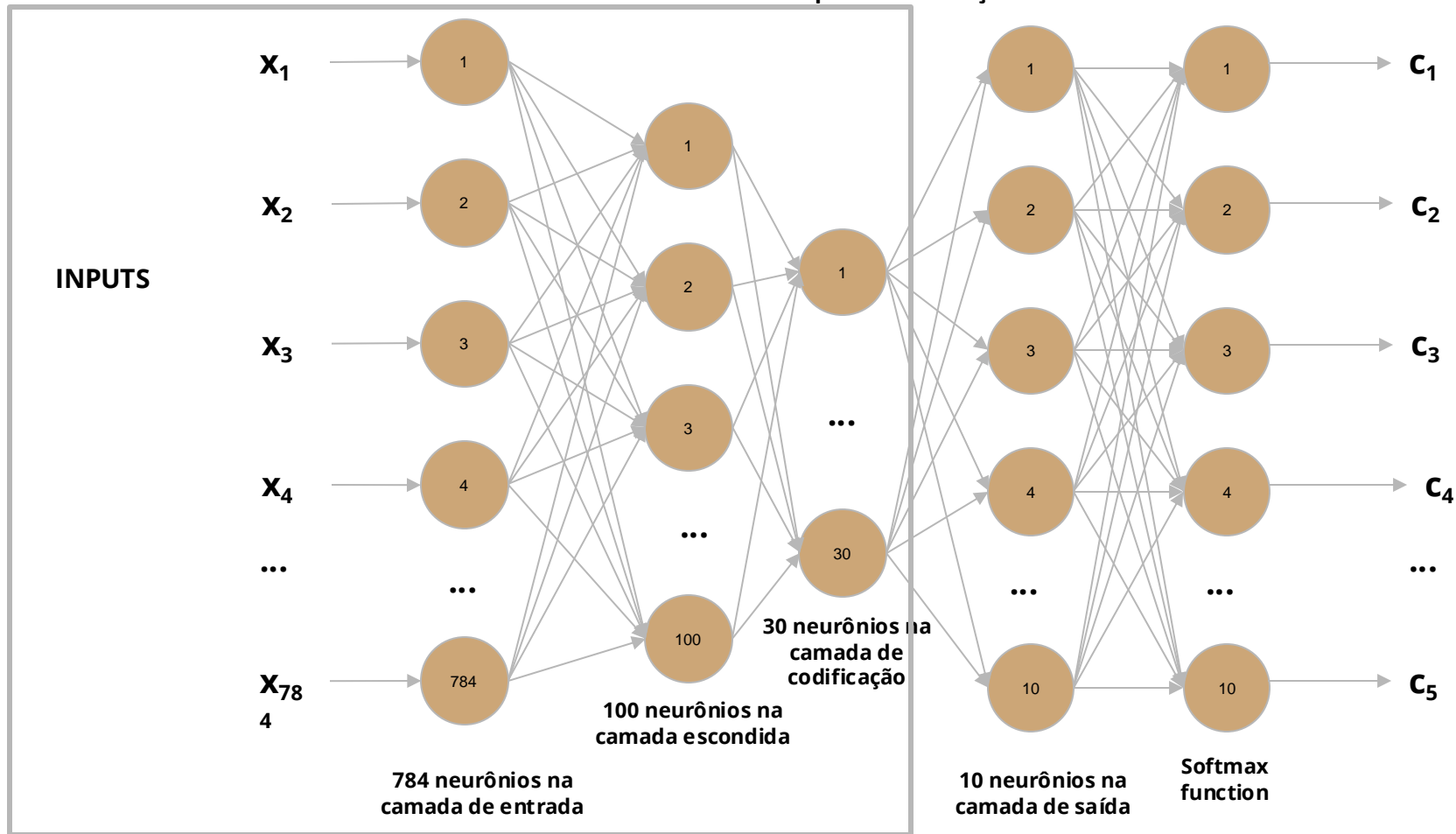
Substitui o decoder por duas novas camadas: uma camada de saída com o número de classes e uma camada softmax



Treinar novamente a rede de forma supervisionada para executar
a tarefa de classificação

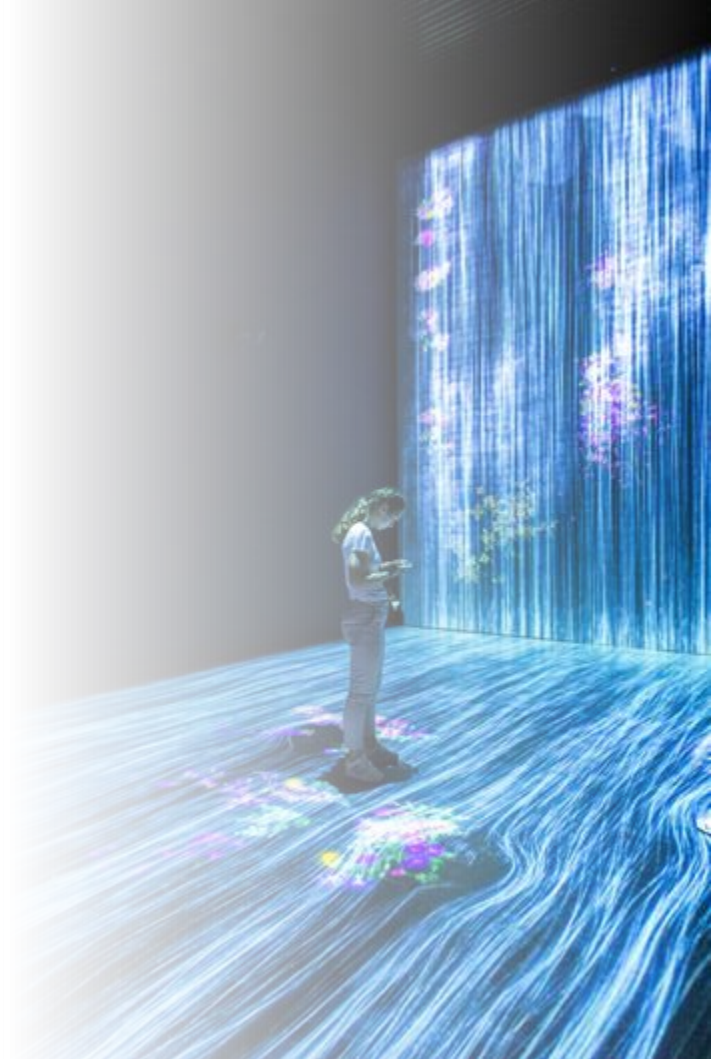


Uma opção é congelar estas camadas iniciais durante o processo de treinamento para classificação



Pré-treinamento não supervisionado

- As etapas deste processo são as seguintes:
 - a. Treinamento de um autoencoder utilizando os dados não anotados (*unlabeled*)
 - b. Reutilize as camadas do *encoder* para criar uma nova rede neural
 - c. Treine esta rede em dados anotados para uma determinada tarefa de classificação. Esta etapa irá precisar de bem menos dados anotados do que o treinamento de um modelo do zero para performar essa mesma tarefa



Transfer Learning

- Este é um exemplo importante sobre transferência de aprendizado. Esta é uma técnica largamente utilizada em Deep Learning.
- A ideia central aqui é aproveitar os pesos da rede de um treinamento feito anteriormente para:
 - minimizar a quantidade de dados necessários para uma determinada tarefa
 - melhorar a acurácia da rede
 - reduzir o tempo de treinamento de uma rede. Na maioria dos casos, é mais caro treinar um modelo do zero.



Transfer Learning

- Aviso importante!!
 - Não estamos explorando aqui todos os aspectos desta técnica. Nosso escopo de atuação aqui são os autoencoders.
 - Transfer learning é uma técnica com dezenas de outras aplicações em deep learning. Vocês vão voltar a ouvir falar sobre esta técnica em disciplinas futuras no curso.
 - A demanda por dados é imensa e este é um grande gargalo em muitos projetos de machine learning. Transfer learning é uma técnica que auxilia muito neste processo.



Transfer Learning - Autoencoders



Transfer Learning - Autoencoders



Tornando treinamento dos
autoencoders mais eficiente

Como treinar os autoencoders de forma mais eficiente?

- O treinamento de autoencoders pode ser bastante custoso. Isso acontece principalmente com os Deep Autoencoders, onde o número de camadas pode ser bem grande
- Uma técnica que pode ser utilizada para tornar estes treinamentos mais eficientes é **“amarrar” os pesos** das camadas de **encoder** com as camadas do **decoder**
- Essa técnica pode ser utilizada quando o **autoencoder é simétrico.**



Como treinar os autoencoders de forma mais eficiente?

- Além de tornar o treinamento do modelo mais eficiente, esta técnica também é capaz de reduzir o número de pesos necessários para representar estas camadas
- Na prática, o que acontece é que estamos “reutilizando” esta camada do *encoder* no *decoder*. Em termos de implementação, elas serão a mesma camada.

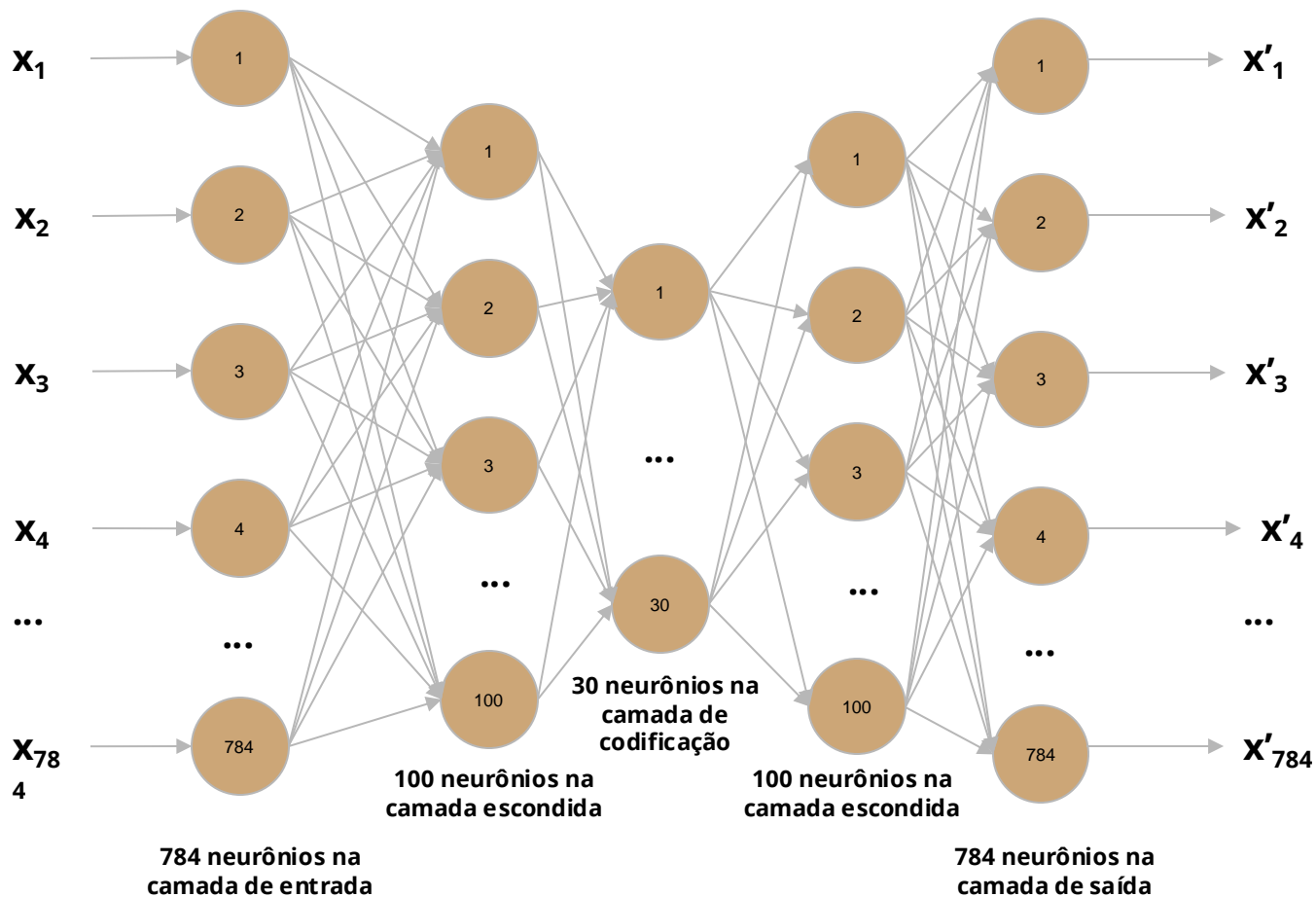


Como treinar os autoencoders de forma mais eficiente?

- Vamos verificar novamente como isso funciona no exemplo de autoencoder que utilizamos para o dataset MNIST:

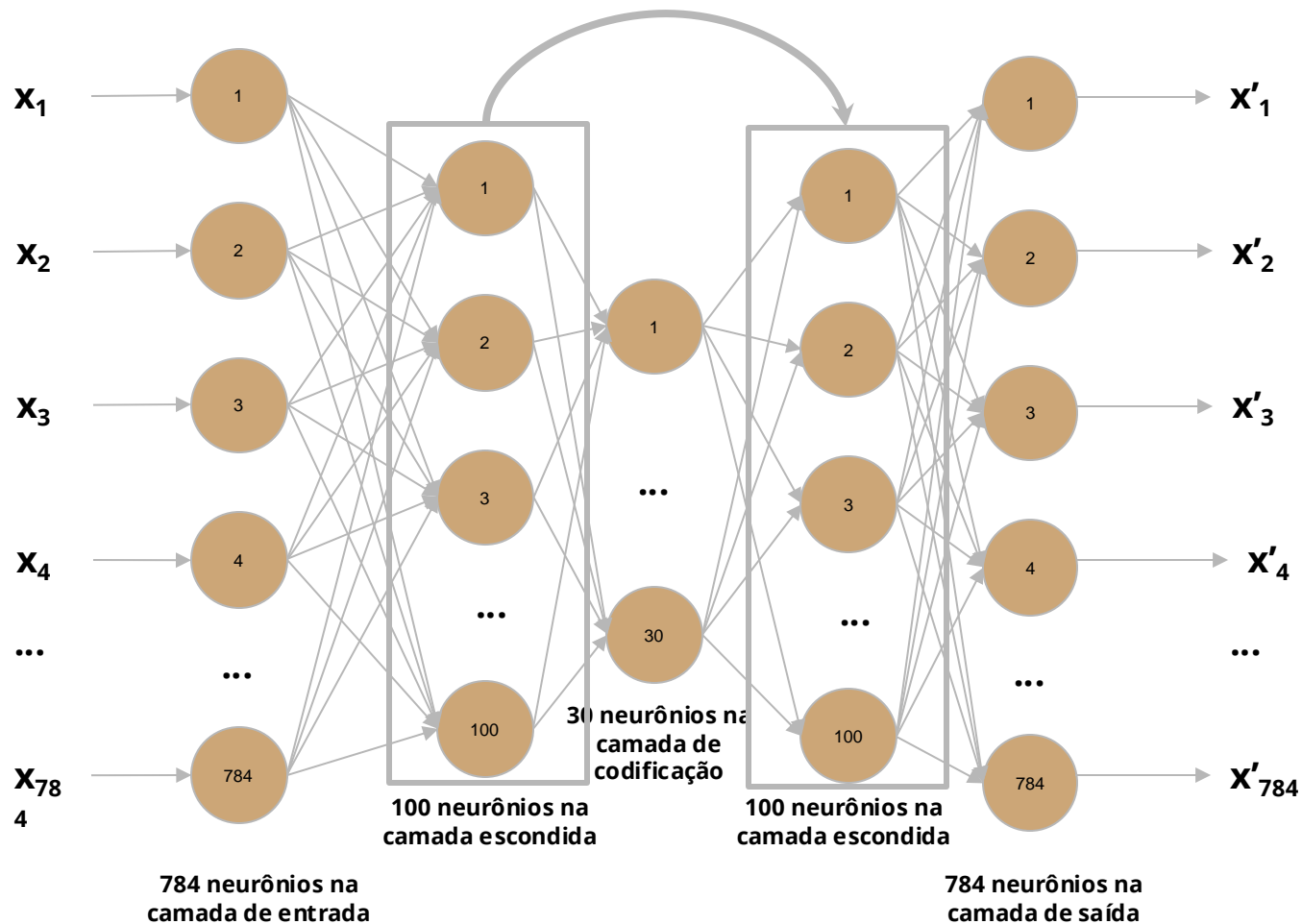


INPUTS



Amarrar estas duas camadas

INPUTS



Copiando os pesos das camadas de
encoder para o decoder

Denoising autoencoders

Denoising autoencoders

- Os autoencoders também podem ser utilizados para remoção de ruído em imagens.
- Para isso, é modificada a imagem de entrada da rede durante o processo de treinamento do autoencoder.
- Vamos ver como ficam os dados de entrada do autoencoder:



Denoising autoencoders

- A implementação deste tipo de autoencoder é simples. Para isso, utilizamos uma camada adicional na entrada do autoencoder.
- Essa nova camada adicional do autoencoder é responsável por aplicar ruído aos dados de entrada.
- Desta forma, o treinamento do autoencoder fará com que a rede neural aprenda a receber uma imagem com ruído e apresentar na saída uma imagem sem ruído.



Denoising autoencoders

- Vamos ver como ficam os dados de entrada do autoencoder:



Autoencoders - Arquiteturas

Autoencoders - Arquiteturas

- Os autoencoders podem ser construídos com diferentes arquiteturas de rede.
- Nós vimos os autoencoders construídos com camadas *fully connected*
- As imagens pequenas podem funcionar com autoencoders deste tipo. Contudo, para imagens maiores é necessário o uso de outros tipos de arquiteturas.



Autoencoders - Arquiteturas

- Para imagens maiores, a ideia é a mesma das redes *fully connected*. Vocês vão construir a arquitetura da rede de modo que a dimensão das imagens seja reduzida no *encoder*. No *decoder* será feito o processo contrário.



Autoencoders - Arquiteturas

- Autoencoders com redes de convolução também são utilizados para segmentação em imagens. Neste caso, a saída do autoencoder é alterada para um mapa onde os pixels de determinado objeto estão marcados.



Autoencoders - Arquiteturas

- Autoencoders também podem ser construídos com redes recorrentes. São redes que têm bastante aplicação em séries **temporais** e para **processamento de texto**.



Trabalho - Autoencoders

Trabalho 3

- Exercício do trabalho:
Autoencoders
- **Tópicos:**
 - Autoencoders
- **Dataset:** Cifar-10
- **Entrega:** TBD



Resumo da Aula de Hoje

- Autoencoders
- Stacked autoencoders



Dúvidas?



Obrigada!