

# Aprendizado de Máquina: RNAs Profundas Avançadas

Prof. Arnaldo Candido Junior  
UTFPR – Medianeira

# Embeddings

- Redes Neurais tem a capacidade de representar seu conhecimento de modo distribuído
- Representação clássica é uma caixa preta:  
**cachorro**, **gato**, **cadeira**
- Representação distribuída: permite comparações  
(ser vivo, late, 4 pernas)  
(ser vivo, mia, 4 pernas)  
(inanimado, silencioso, 4 pernas)
  - Quem é mais parecido com quem?

# Embeddings <sub>(2)</sub>

- Essa representação distribuída é chamada de embeddings ou encodings ou características
- Exemplo: análise em redes convolucionais supervisionados para análise de imagens
  - A seguir

# Embeddings <sub>(3)</sub>

- Suponha que a rede gerou representação com informações como:
  - Tamanho do animal
  - Comprimento do focinho
  - Distância entre os olhos
  - Tipo de pelagem

# Embeddings <sub>(4)</sub>

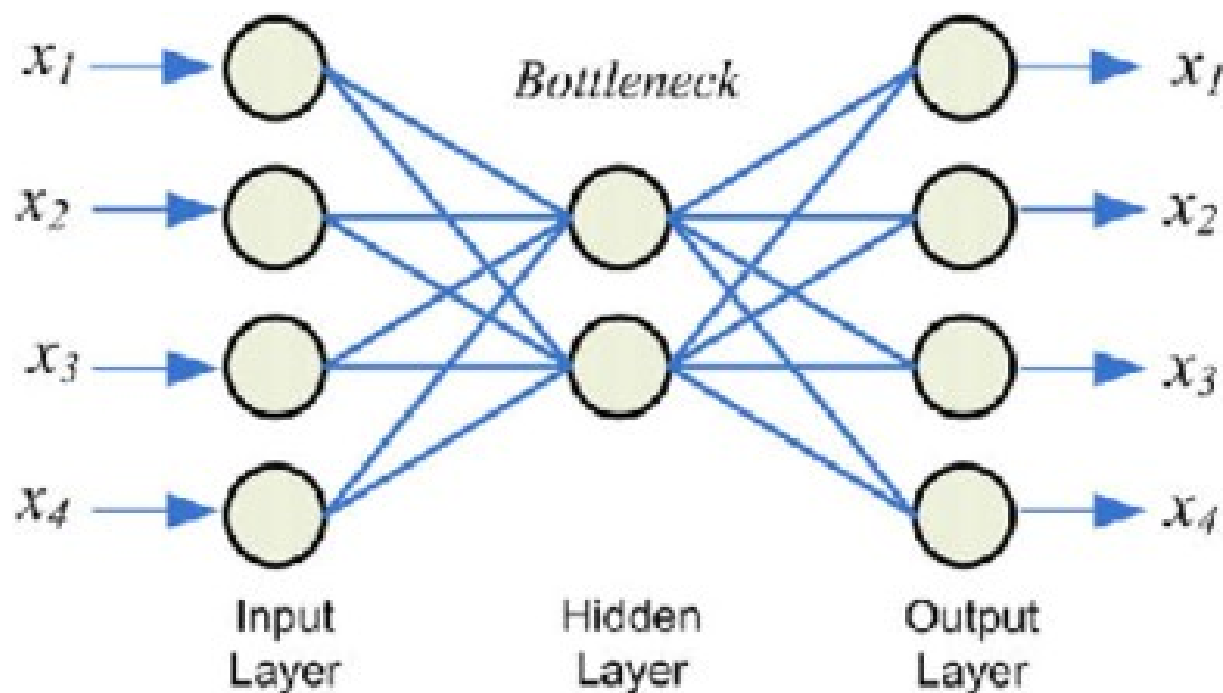
- Resultado desejado:
  - Embeddings de gatos são muito semelhantes entre si (ou camada de saída não conseguiria classificar gatos)
  - Embeddings de gatos são muito distintos de embeddings de cães (ou a camada de saída confundiria gatos com cães)
  - Mesmo raciocínio para outros mamíferos

# Embeddings (5)

- Embeddings são capazes de representar animais não vistos em treinamento
  - Lince: representação boa pois pode usar as características aprendidas para gato para representá-lo
  - Jacaré/Gaivota: representação intermediária. Informações como focinho e pelagem
  - Peixe/minhoca: representação começa a ficar pobre

# Autoencoders

- São redes que tentam reconstruir a entrada na saída
- Exemplo: autoencoder raso



# Autoencoders <sub>(2)</sub>

- Normalmente, a camada escondida é menor que a entrada
- Rede é incentivada a extrair informações úteis da entrada de modo compactado
- Funciona como um algoritmo de compactação com perdas

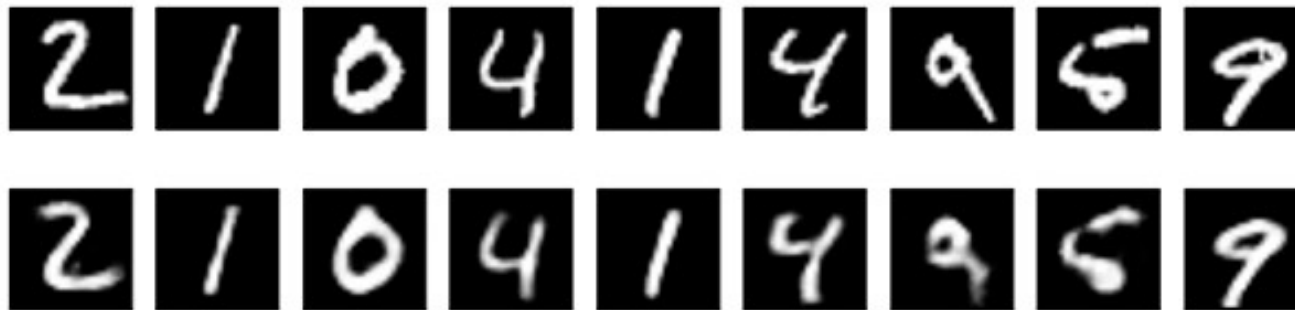


# Autoencoders <sub>(3)</sub>

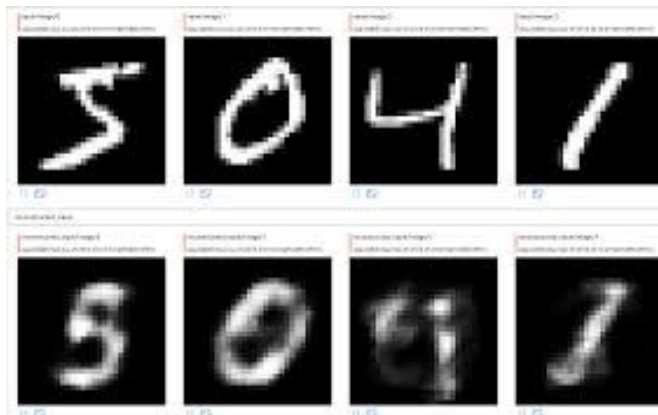
- São especializados em um tipo de entrada (ex.: números Mnist)
- Possuem variantes profundos, recorrentes e convolucionais disponíveis
- Bastante úteis para extrair embeddings

# Autoencoders <sup>(3)</sup>

- Boas reconstruções:

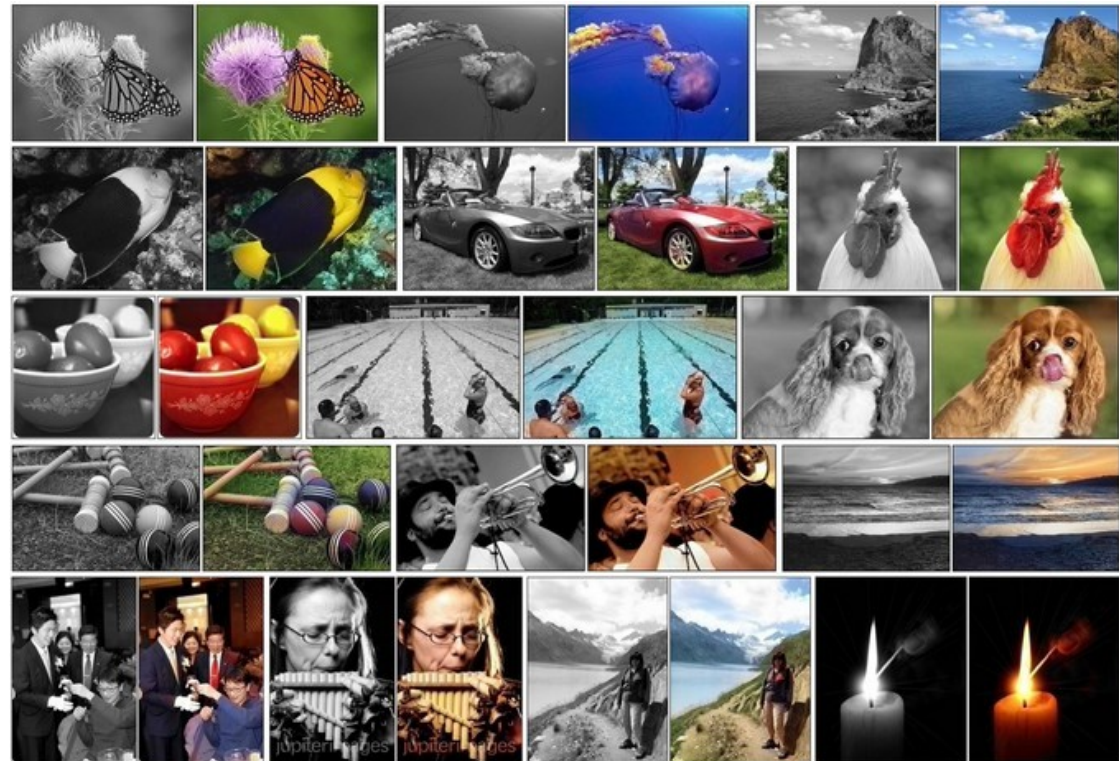


- Reconstrução menos eficiente:



# Autoencoders (4)

- São a base de muitas redes avançadas nas quais a saída é uma versão modificada da entrada
- Exemplo: colorização de fotos



# Word Embeddings

- Base para tradução e processamento de língua natural
- Palavras são tradicionalmente representadas por vetores 1-hot
  - Muito esparsos → maldição da dimensionalidade
- Embeddings são representações densas de baixa dimensionalidade (50 até 300)

# Word Embeddings <sub>(2)</sub>

- Base para tradução e processamento de língua natural
- Palavras são tradicionalmente representadas por vetores 1-hot
  - Muito esparsos → maldição da dimensionalidade
- Embeddings são representações densas de baixa dimensionalidade (50 até 300)

# Word Embeddings <sub>(3)</sub>

- Exemplo intuitivo:
  - Cão:  $(-1, 0.5, 3)$
  - Cachorro:  $(-0.9, 0.4, 3.1)$
  - Gato:  $(-2, 0.5, 2.9)$
  - Cadeia:  $(5, -3, 0)$
- Quem é mais similar a quem?

# Word Embeddings <sub>(4)</sub>

- Word embeddings são treinados com frases como “o gato pulou no colchão” em que uma das palavras é removida
- Rede deve adivinhar palavra removida
  - Entrada:  $1\text{hot}(\text{o}) + 1\text{hot}(\text{gato}) + 1\text{hot}(\text{no}) + 1\text{hot}(\text{colchão})$
  - Camada oculta: cria o embedding
  - Saída:  $1\text{hot}(\text{pulou})$

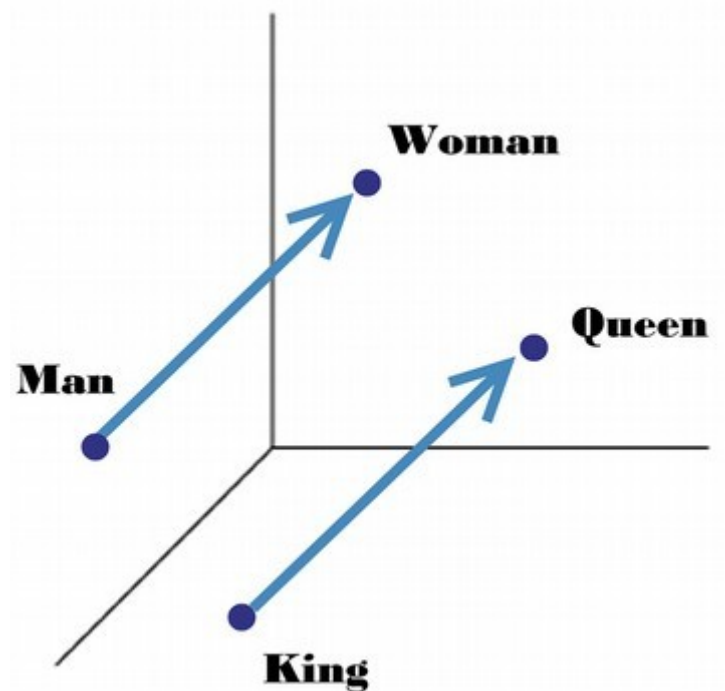
# Word Embeddings (5)

- Cão e cachorro tem praticamente as mesmas palavras vizinhas (latir, ração, etc)
  - Recebem praticamente os mesmos ajustes de pesos
  - Ficam muito parecidas
- $v = \text{emb}(\text{rei}) - \text{emb}(\text{homem}) + \text{emb}(\text{mulher})$ 
  - Existiria algum embedding no dataset semelhante ao resultado em  $v$ ?



# Word-embedding <sub>(6)</sub>

- $\text{emb}(\text{rei}) - \text{emb}(\text{homem}) \approx \text{emb}(\text{rainha}) - \text{emb}(\text{mulher})$
- Lê-se: rei está para homem assim como rainha está para mulher



# Word-embedding <sub>(7)</sub>

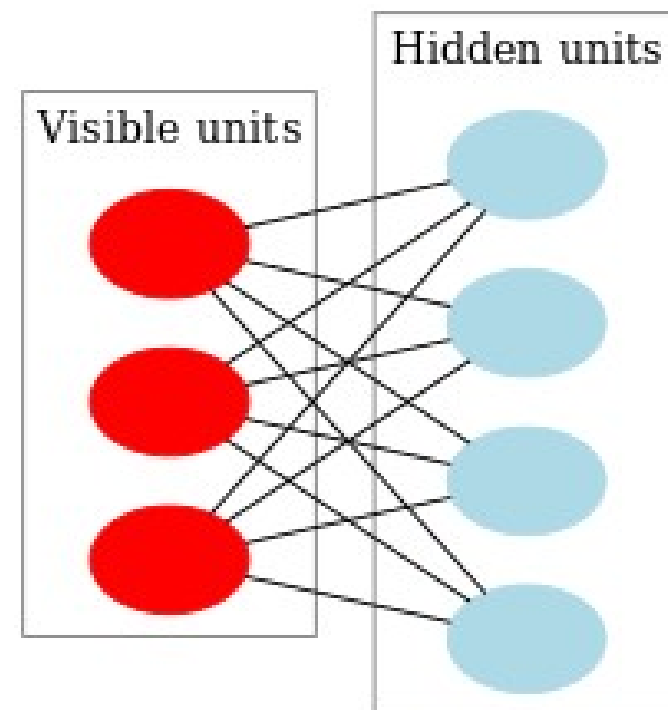
- Entendendo o comportamento visto
  - Contexto “monarquia”: puxa rei e rainha em uma direção do espaço
  - Contextos “humano”, “feminino” e “masculino” surtem efeito parecido nos embeddings do exemplo
  - Esse “cabo de guerra” entre os contextos fazem os embeddings assumirem as posições do exemplo

# Máquinas de Boltzmann

- São um tipo de **rede generativa**.
- Classificação está associados a probabilidades **a posteriori**
  - $P(\text{doença} \mid \text{sintomas})$ ;  $P(\text{gato} \mid \text{foto})$
- Redes generativas estão associadas a probabilidade **a priori**
  - $P(\text{gato})$
  - Gera uma nova foto de um gato

# Máquinas de Boltzmann (2)

- Máquina Restrita de Boltzmann
  - Tem uma camada que é simultaneamente entrada e saída
  - Valor aleatório é inserido na camada de E/S
  - Valor é propagado várias vezes entre a camada E/S e a camada oculta



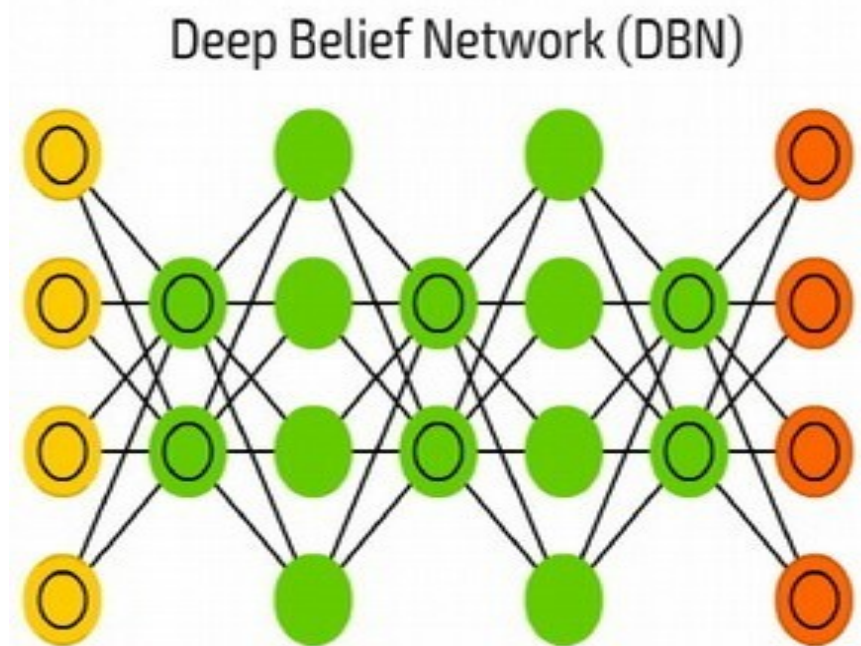
# Máquinas de Boltzmann <sup>(3)</sup>

- Saída converge para uma nova instância para o problema que foi treinado
- Exemplo: gerar novos dígitos no estilo Mnist



# Máquinas de Boltzman (4)

- **Deep Belief Network:**
  - Pode ser pensada como uma variante profunda da Máquina Restrita de Boltzmann
  - Diferente da rede convolucional, não constrói uma hierarquia de conceitos

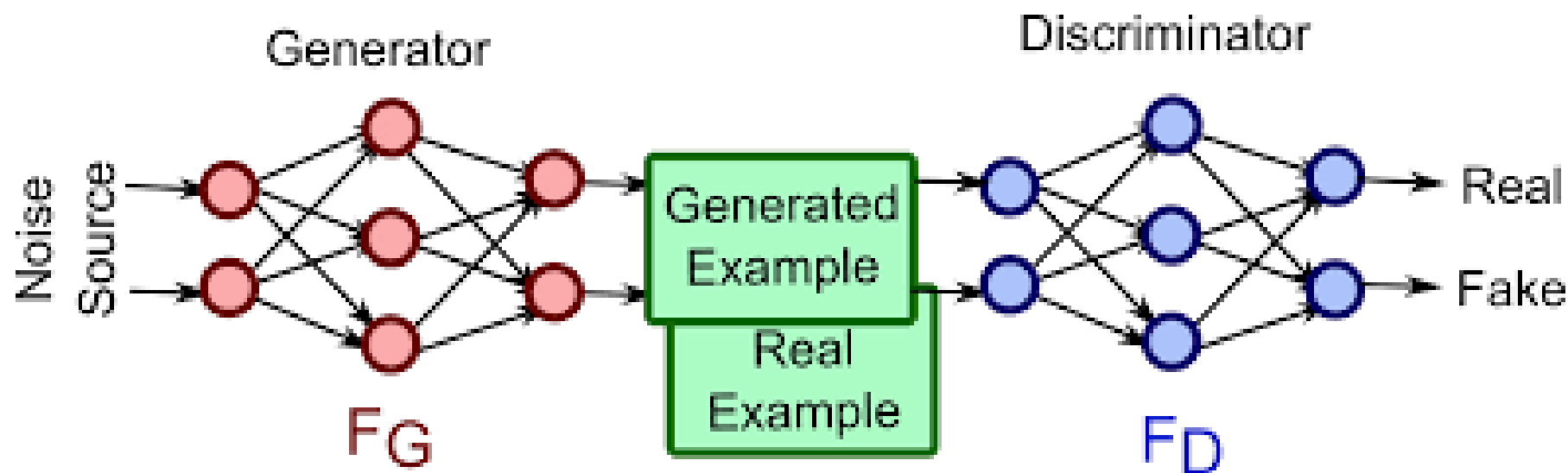


# Redes Generativas Adversariais

- São outro tipo de rede generativa:
  - Duas sub-redes que competem dentro de uma rede maior
  - A primeira tenta criar instâncias “falsas” similares ao dataset para enganar a segunda
  - A segunda tenta distinguir entre instâncias verdadeiras e falsas
  - A medida que competem, as falsificações vão ficando melhor e melhor

# Redes Generativas Adversariais <sup>(2)</sup>

- Rede 1: minimiza a taxa de acerto da saída (entrada é vetor aleatório)
- Rede 2: maximiza a taxa





# Redes Generativas Adversariais <sup>(3)</sup>

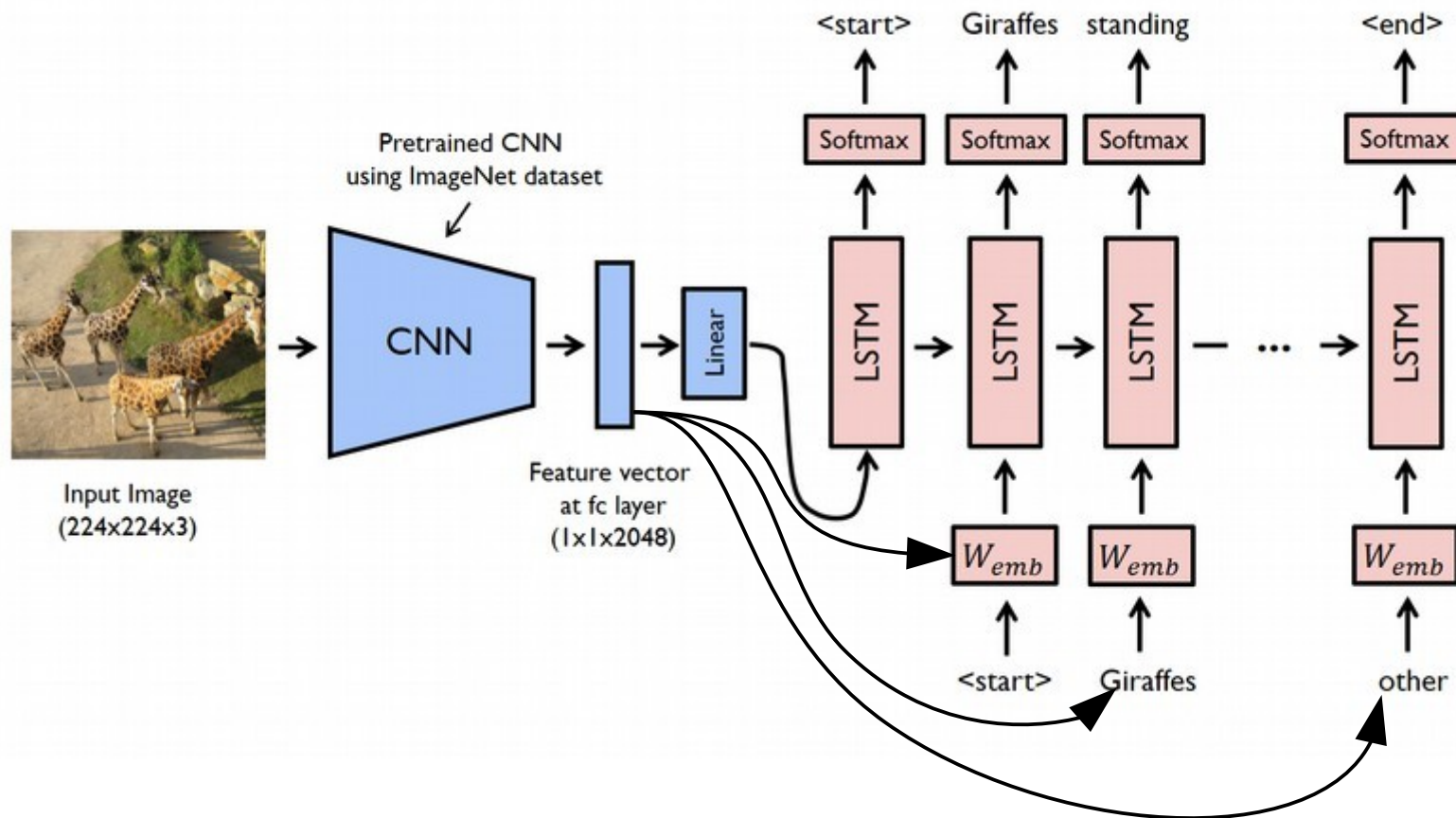
- Estão revolucionando a área de geração de imagens e vídeos
- Exemplo: celebridades falsas



# Redes com atenção

- Conseguem se focar em determinadas partes da entrada para tomar uma decisão
  - Importante para tradução neural e em diversos tipos de redes convolucionais
  - Veremos um exemplo para geração de legendas de imagens
    - Pode ser pensado como um tradutor cuja língua fonte é uma imagem e a língua alvo é o inglês

# Redes com atenção (2)



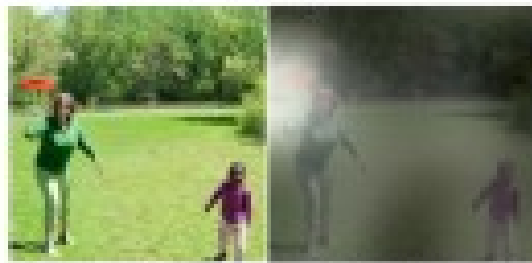
# Redes com atenção <sup>(3)</sup>

- No exemplo, para gerar uma palavra a rede olha:
  - A anterior produzida
  - A representação de embedding da entrada
  - Com base nisso, ela decide se focar em uma parte da entrada relevante para a próxima palavra

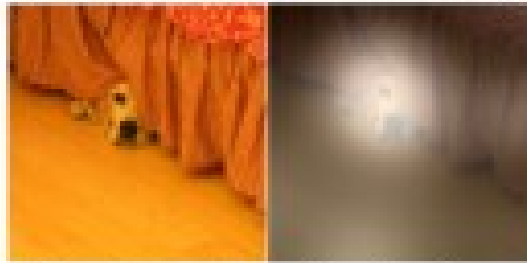
# Redes com atenção <sup>(4)</sup>

- Existem vários tipos de atenção, a seguir, um exemplo de soft attention

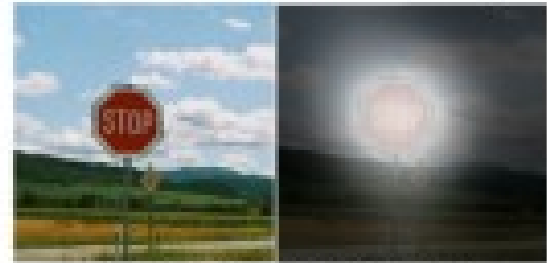
Figure 3. Examples of attending to the correct object (white indicates the attended regions, underlines indicated the corresponding word)



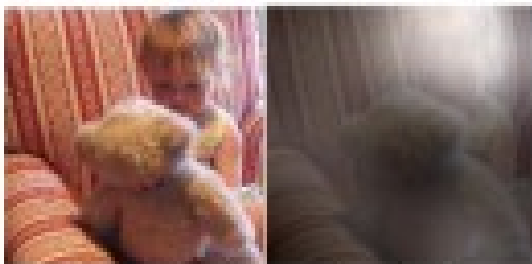
A woman is throwing a frisbee in a park.



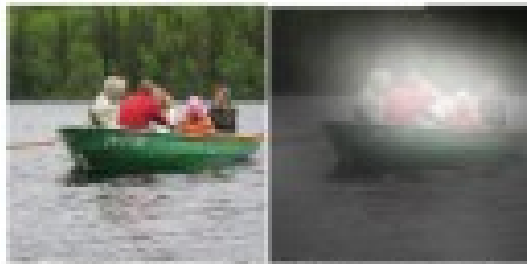
A dog is standing on a hardwood floor.



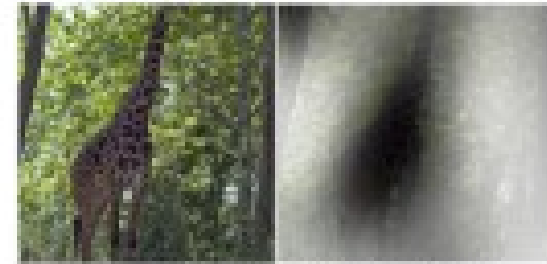
A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

# Aprendizado por Reforço

- Resumo: o aprendizado por reforço usa uma tabela de estados
- Na tabela:  
verde = pote de ouro;  
vermelho = boma;  
branco vazio = bloqueio
- Demais ladrilhos: estados possíveis em que o personagem pode estar

<div>- 0.08</div> <div>- 0.08</div> <div>0.88</div>	<div>- 0.04</div> <div>- 0.07</div> <div>0.92</div>	<div>- 0.04</div> <div>- 0.04</div> <div>0.96</div>	1.00
<div>- 0.09</div> <div>0.84</div> <div>- 0.10</div>	<div>- 0.04</div> <div>- 0.10</div> <div>- 0.10</div>	<div>- 0.03</div> <div>0.51</div> <div>- 0.05</div>	- 0.88
<div>- 0.10</div> <div>- 0.10</div> <div>0.79</div>	<div>- 0.10</div> <div>- 0.12</div> <div>- 0.10</div>	<div>- 0.05</div> <div>0.08</div> <div>- 0.09</div>	- 0.27
<div>- 0.12</div> <div>- 0.12</div> <div>- 0.12</div>	<div>- 0.10</div> <div>- 0.10</div> <div>- 0.09</div>	<div>- 0.08</div> <div>- 0.08</div> <div>- 0.08</div>	- 0.08

# Aprendizado por Reforço <sub>(2)</sub>

- Utilidades dos estados próximos ao final são aprendidas rapidamente
- Utilizados dos estados iniciais são aprendidas devagar
  - Existem muitos mais estados próximos iniciais do que estados próximos ao fim de jogo
- Estados aproximados / número de partidas aproximadas: Xadrez:  $10^{123}$ ; Go:  $10^{360}$ ; jogos eletrônicos: difícil de estimar

# Aprendizado por Reforço <sup>(3)</sup>

- Solução: modificar o algoritmo de aprendizado por reforço para usar uma RNA
  - RNA olha estado e diz utilidade
  - Não precisa memorizar todos os estados
  - Estados parecidos tem utilidade parecidas
- Tendência: agentes curiosidade
  - Especificação matemática que força o agente a explorar mais os estados em que ele tem dúvida do resultado



# Aprendizado por Reforço <sub>(4)</sub>

- Redes neurais estão revolucionando a área de aprendizado por reforço
- Jogos antes impensáveis de serem aprendidos estão sendo jogados pelas redes
  - Diversos de Atari
  - Dota
  - StarCraft 2