Advanced Institute for Artificial Intelligence

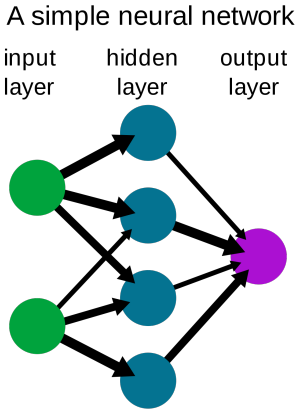
Processamento de Textos (NLP)

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Treinando Redes Neurais

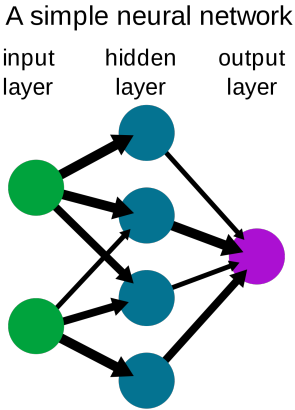
Agora vocês sabem como treinar uma rede neural:



- Treinar a Rede Neural requer especificar apenas uma função de custo, sua arquitetura e quais são suas entradas
- Entretanto, nem toda aplicação se traduz facilmente em um conjunto de variáveis numéricas

Como passar um texto para uma RN?

Estou feliz em porque aprendi IA



Primeira Tentativa

Representar Texto da forma como ele é representado internamente no computador

ASCII printable characters					
32	space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o

- A Tabela ascii descreve números naturais que representam caracteres de texto.
- É possível representar cada caracter em uma frase através de seu código ascii correspondente
- Assim teríamos códigos numéricos que poderiam ser utilizados em uma rede neural!

- A Tabela ascii descreve números naturais que representam caracteres de texto.
- É possível representar cada caracter em uma frase através de seu código ascii correspondente
- Assim teríamos códigos numéricos que poderiam ser utilizados em uma rede neural!

Estou feliz porque aprendi IA $\rightarrow$

69 115 116 111 117 32 102 101 108 105 122
32 112 111 114 113 117 101 32 97 112 114
101 110 100 105 32 73 65

Problema

Presença (e até número) de letras não ajuda a desvendar sentido da frase

$$\textit{Inseto} \rightarrow 105\ 110\ 115\ 101\ 116\ 111$$

$$\textit{Isento} \rightarrow 105\ 115\ 101\ 110\ 116\ 111$$

- Entretanto, **palavras** raramente têm múltiplos significados:

Estou *feliz* porque aprendi IA

Como Processar Palavras?

- Podemos atributir um código arbitrário para cada palavra encontrada
- Este vocabulário é utilizado para fazer correspondência entre frases

Estou feliz porque aprendi IA

[1, 2, 3, 4, 5]

vocabulário = {estou: 1, feliz: 2, porque: 3, aprendi: 4, ia: 5}

Estou feliz porque terminou a quarentena

[1, 2, 3, 6, 7, 8]

vocabulário = {estou: 1, feliz: 2, porque: 3, aprendi: 4, ia: 5, terminou: 6, a: 7, quarentena: 8}

Estou feliz porque aprendi IA $\rightarrow$ [1, 2, 3, 4, 5]

Estou feliz porque terminou a quarentena $\rightarrow$ [1, 2, 3, 6, 7, 8]

- Podemos recuperar o vocabulário do conjunto de treinamento
- Vocabulários extensos podem resultar em uma acurácia maior, mas aumentam exponencialmente a "dificuldade" de treinar o modelo
- Um número predefinido de palavras pode ser considerado para o vocabulário
  - Palavras menos frequentes podem ser descartadas

- Palavras fora do vocabulário são descartadas

Quero ter um *Agumon* agora

vocabulário = {quero: 1, ter: 2, um: 3, agora: 4}

→ [1, 2, 3, 4]

- Ou podem ser substituídas por um símbolo padrão

→ [1, 2, 3, <OOV>, 4]

```
from tensorflow.keras.preprocessing.text import Tokenizer

frases = [ 'Estou feliz porque aprendi IA',
           'Estou triste, não posso sair de casa'
]

tokenizer = Tokenizer(num_words = 100)
tokenizer.fit_on_texts(frases)
word_index = tokenizer.word_index
print(tokenizer.word_index)
```

- A classe $Tokenizer$ prepara um vocabulário a partir de um *corpus* de sentenças.
- O vocabulário resultante está ordenado por frequência de palavras.
- Novas sequências podem ser codificadas usando o objeto tokenizer.

Alimentando uma RN com textos

- Agora sabemos como processar o texto em um formato mais amigável para uma rede neural

Alimentando uma RN com textos

- Agora sabemos como processar o texto em um formato mais amigável para uma rede neural
- Mas ainda há um problema.....

Estou feliz porque aprendi IA $\rightarrow$ [1, 2, 3, 4, 5]

Estou triste, não posso sair de casa $\rightarrow$ [1, 6, 7, 8, 9, 10]

$\text{len}([1, 2, 3, 4, 5]) = 5$

$\text{len}([1, 6, 7, 8, 9, 10]) = 6$

- Redes neurais possuem um tamanho fixo de entrada
- Textos são arbitrariamente grandes
- Como resolver isso?

Pré-processando Sequências codificadas

- Devemos definir um tamanho máximo possível de texto relevante para nosso domínio
- Por exemplo:
  - 280: Se a tarefa classifica textos provenientes do Twitter (tamanho máximo de postagem)
  - 800: Review de filmes
  - 20000: Tradução de documentos

- Depois de definido o tamanho máximo, normalizamos todos os textos (treinamento e teste)
- Utilizando uma técnica chamada *Padding*

Padding

- Padding consiste em adicionar zeros no início (ou final) da frase até que atinja o tamanho desejado
- [1,2,3,4], max_len=10 $\rightarrow$ [1,2,3,4,0,0,0,0,0,0]

```
from tensorflow.keras.preprocessing.sequence import pad_sequences

sequences = np.array([1,2,3,4], [1,2], [1,2,3,4,5,6])
pad_sequences(sequences, maxlen=5, padding='post', truncating='post')

[
  [1,2,3,4,0],
  [1,2,0,0,0],
  [1,2,3,4,5]
]
```

Agora já sabemos como processar o texto!

1. A partir do conjunto de treinamento, definir o vocabulário e codificá-lo
2. Codificar todas as sentenças de acordo com o vocabulário
3. Executar *padding* das sentenças
4. Treinar o modelo

Obs: O vocabulário e configurações de padding precisam ser salvos para que o modelo seja utilizado para predições.

Embeddings

Mas ainda resta um problema!

- Palavras semelhantes ficam arbitrariamente afastadas no conjunto de treinamento
- A tarefa de classificação é muito difícil

Que filme horrível $\rightarrow$ [238, 1, 170]

Que filme terrível $\rightarrow$ [238, 1, 5]

Que filme horrível $\rightarrow$ [238, 1, 170]

Que filme terrível $\rightarrow$ [238, 1, 5]

- Palavras que exercem função semelhante na frase ficam arbitrariamente afastadas
- Generalização se torna praticamente impossível

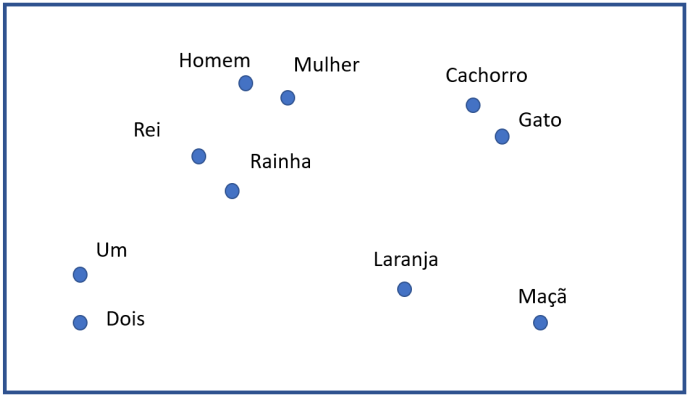
Word Embeddings

- Como fazer com que o modelo entenda *analogias*: homem $\times$ mulher, rei $\times$ rainha, maçã $\times$ laranja
- O segredo está na representação!

- Embeddings transformam a frase codificada em um vetor de maior dimensão, em que palavras semelhantes ficarão próximas
- Embeddings: $\mathbb{N}^{length} \rightarrow \mathbb{R}^{e \times length}$

	Homem (251)	Mulher (1)	Rei (7)	Rainha (174)	Maçã (14)	Laranja (110)
Gênero	-1	1	-0.95	0.97	0	0.01
Realeza	0.01	0.02	0.93	0.95	-0.01	0
Idade	0.03	0.02	0.7	0.69	0.03	-0.02

Visualizando Embeddings



Word Embeddings

- Utilizando embeddings é possível generalizar palavras de acordo com sua utilidade para a tarefa
- Também é possível utilizar embeddings gerados por outros grupos a partir de uma quantidade massiva de textos

```
from tensorflow.keras.layers import Embedding

embedding_dim = 16

model = Sequential()
model.add(Embedding(vocab_size, embedding_dim, input_length=maxlen))
```