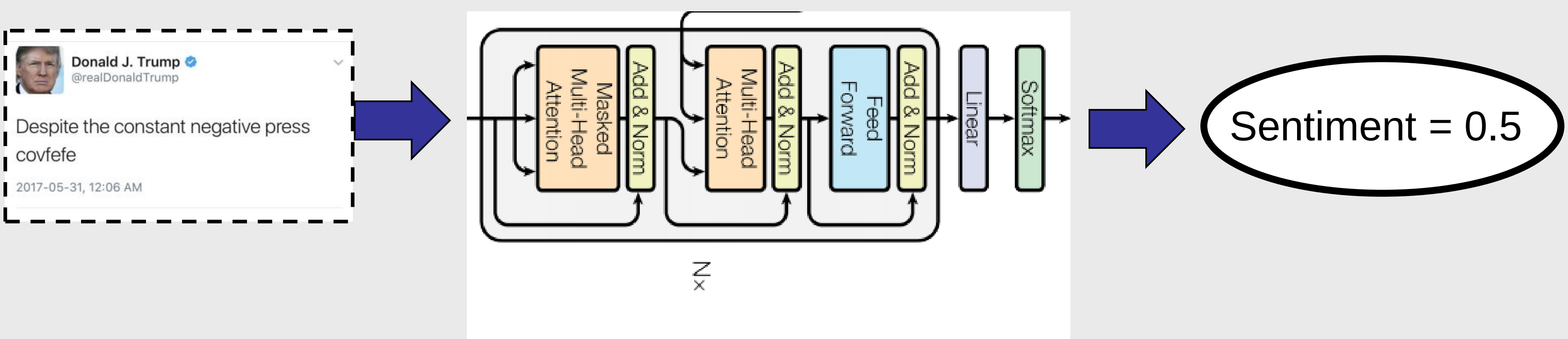
Sentiment Analysis with Neural Network Transformers



Sentiment

Sentiment

- **Tweet 1: My birthday cake was awful**

Sentiment

- Tweet 1: My birthday cake was awful

- Tweet 2: My birthday cake was great

Sentiment

- Sentiment is conveyed by specific words

Sentiment

- Sentiment is conveyed by specific words

- Maybe we could use a word frequency approach to measure sentiment

Sentiment

- Sentiment is conveyed by specific words
- Maybe we could use a word frequency approach to measure sentiment
- Naïve Bayes classifier – measure sentiment using term frequency embeddings

Sentiment and Context

Sentiment and Context

- Tweet 1: My birthday cake was great, if you want my honest opinion

Sentiment and Context

- **Tweet 1: My birthday cake was great, if you want my honest opinion**

- **Tweet 2: My birthday cake was great, if you want me to get diabetes**

Sentiment and Context

- Tweet 1: My birthday cake was great, if you want my honest opinion

- Tweet 2: My birthday cake was great, if you want me to get diabetes

Sentiment and Context

- **Tweet 1: My birthday cake was great, if you want my honest opinion**

- **Tweet 2: My birthday cake was great, if you want me to get diabetes**

Sentiment and Context

- Sentiment is conveyed by specific words

Sentiment and Context

- Sentiment is conveyed by specific words
- We also need to know the context of the words

Sentiment and Context

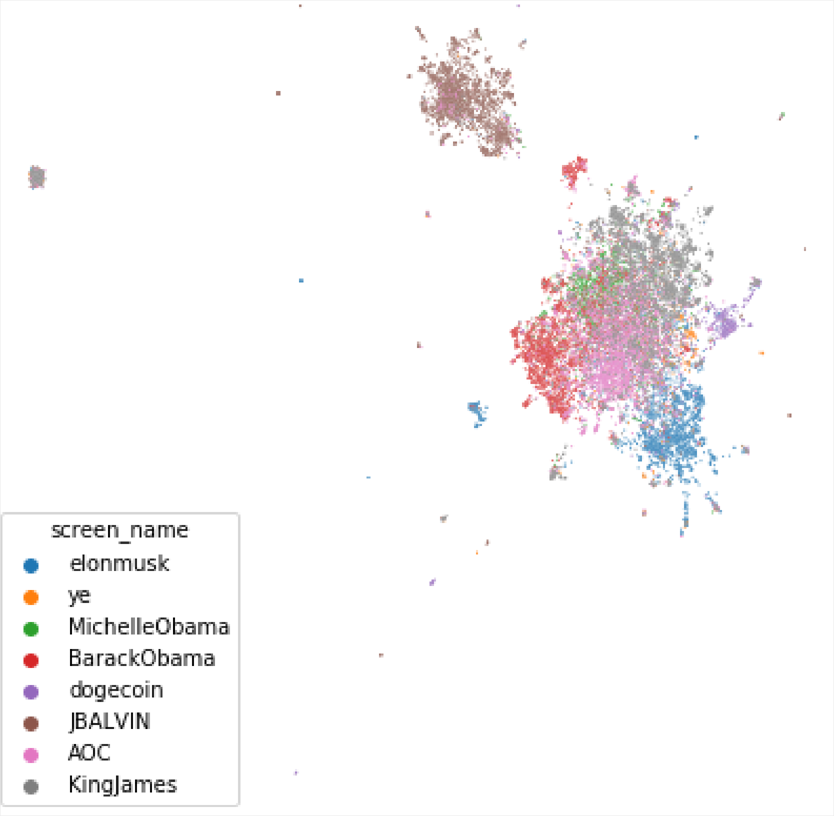
- **Sentiment is conveyed by specific words**

- **We also need to know the context of the words**

- **Context = which words pay attention to which words**

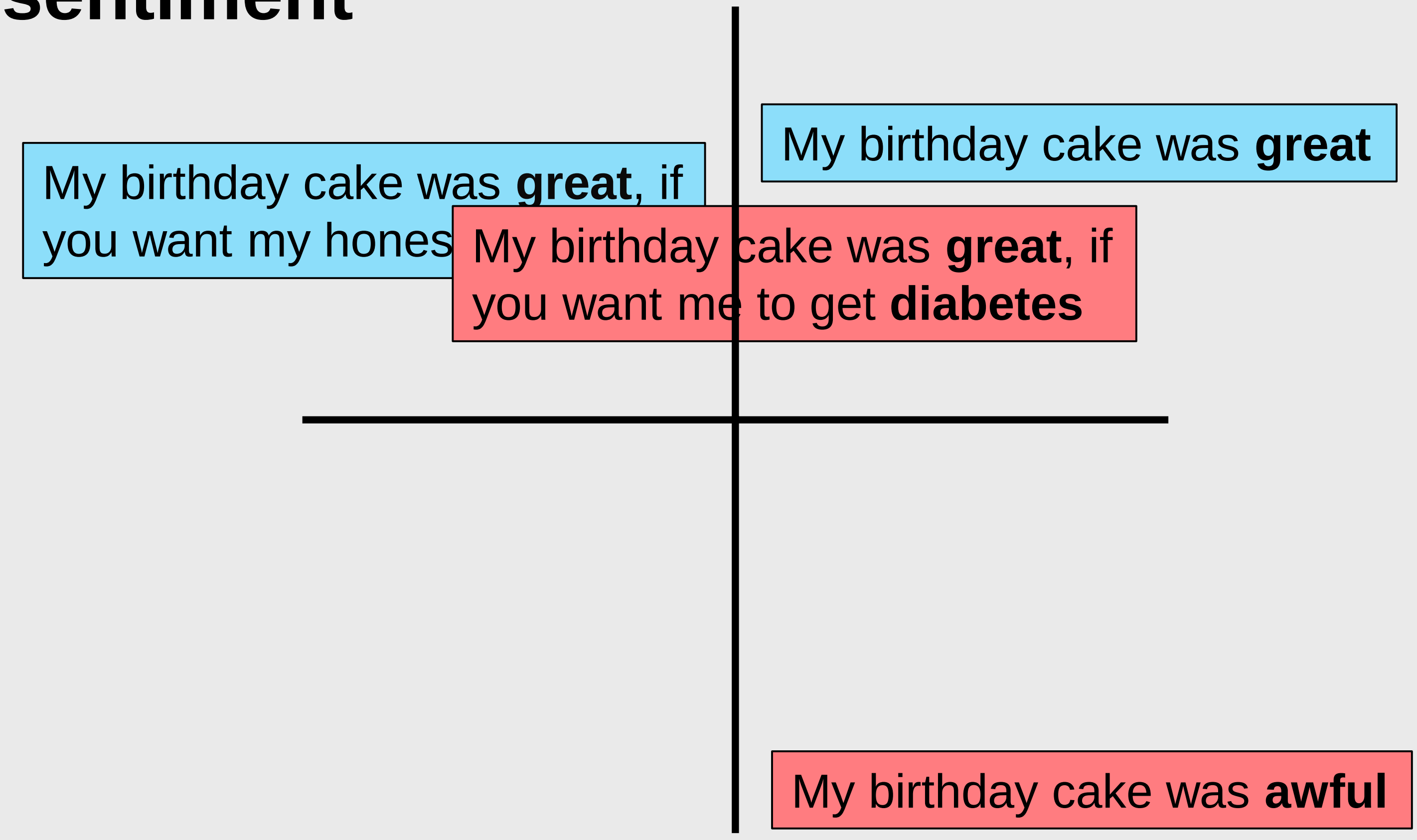
Embeddings

- We have seen how embeddings make clustering text easy



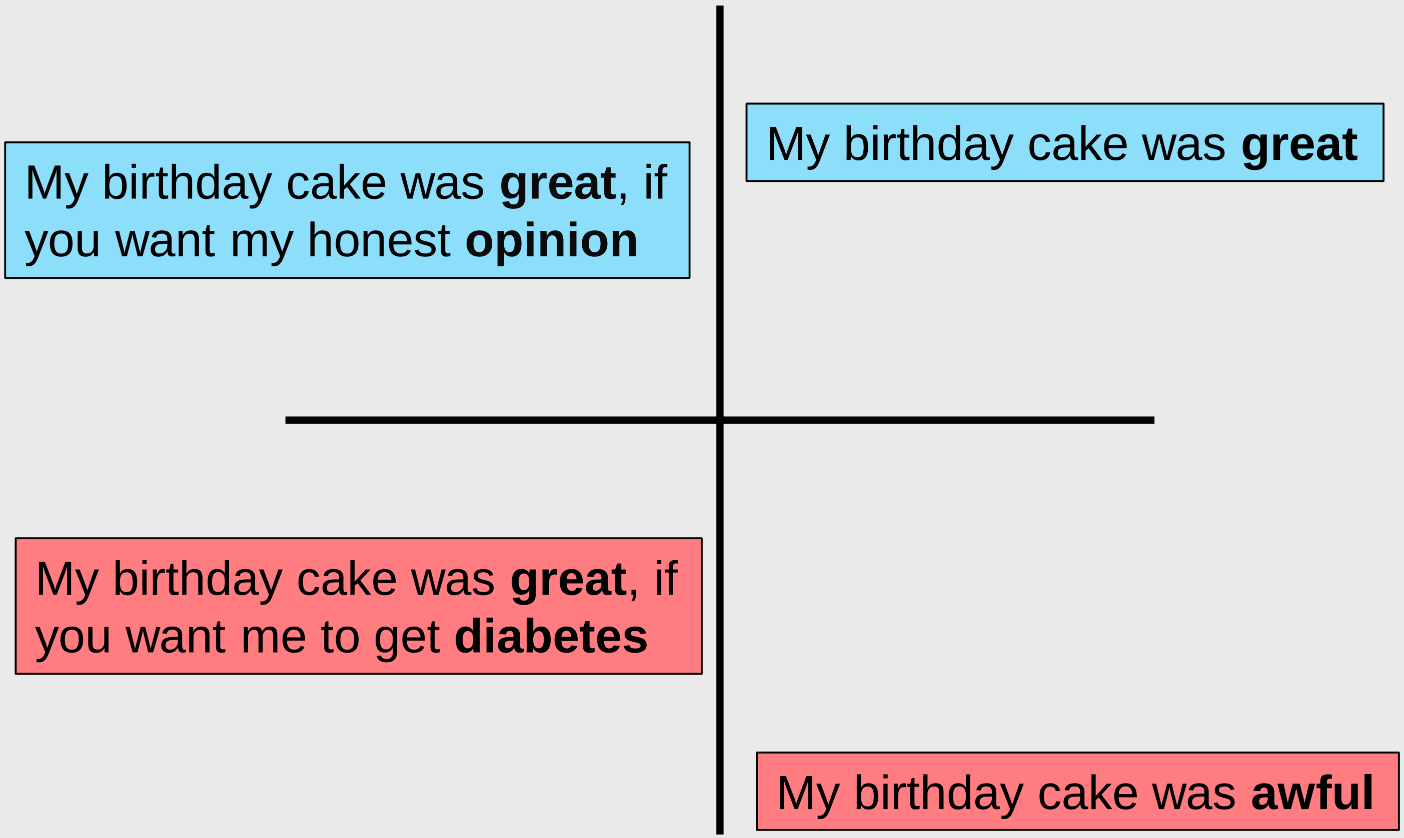
Context Dependent Embeddings

- **A clustering type of embedding may cluster tweets with similar words, but different sentiment**



Context Dependent Embeddings

- **Context dependent embedding can cluster by sentiment**



Context Dependent Embeddings

Context Dependent Embeddings

- We need a model that allows words in a sentence to pay “attention” to other words

Context Dependent Embeddings

- We need a model that allows words in a sentence to pay “attention” to other words

- Words can pay attention in different ways

Context Dependent Embeddings

- **We need a model that allows words in a sentence to pay “attention” to other words**

- **Words can pay attention in different ways**

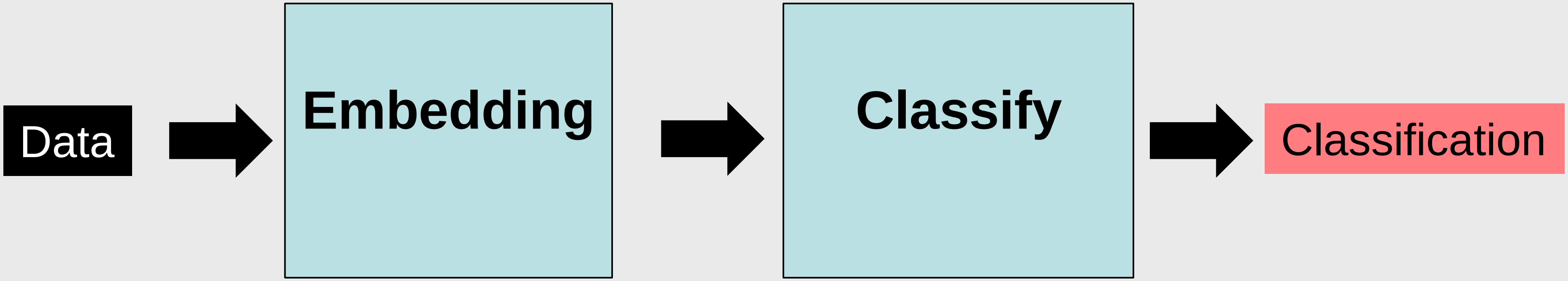
- **We can choose the type of “attention” that captures sentiment**

Context Dependent Embeddings

- **We need a model that allows words in a sentence to pay "attention" to other words**
- **Words can pay attention in different ways**
- **We can choose the type of "attention" that captures sentiment**
- **Solution: Neural Network Transformers**

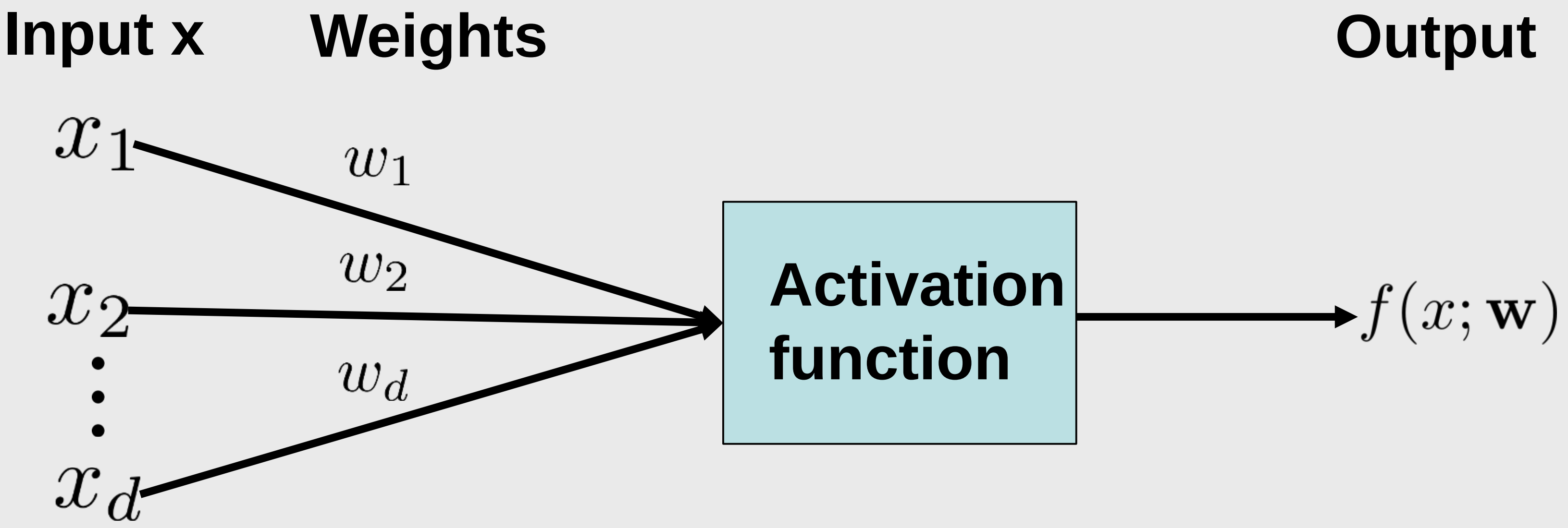
Neural Networks

- Neural networks let you learn very complex embeddings
- You can classify data using these embeddings



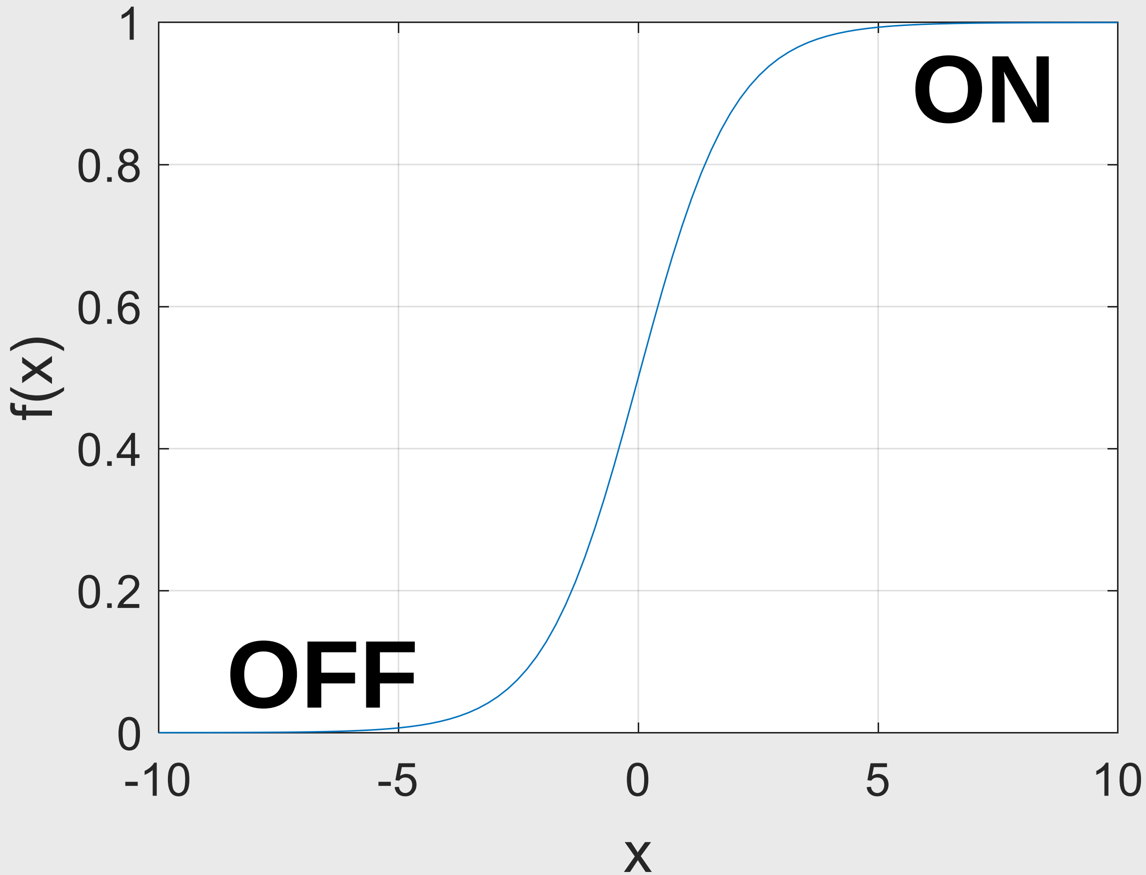
Neurons

- Neurons are the core building block of a neural network is the neuron



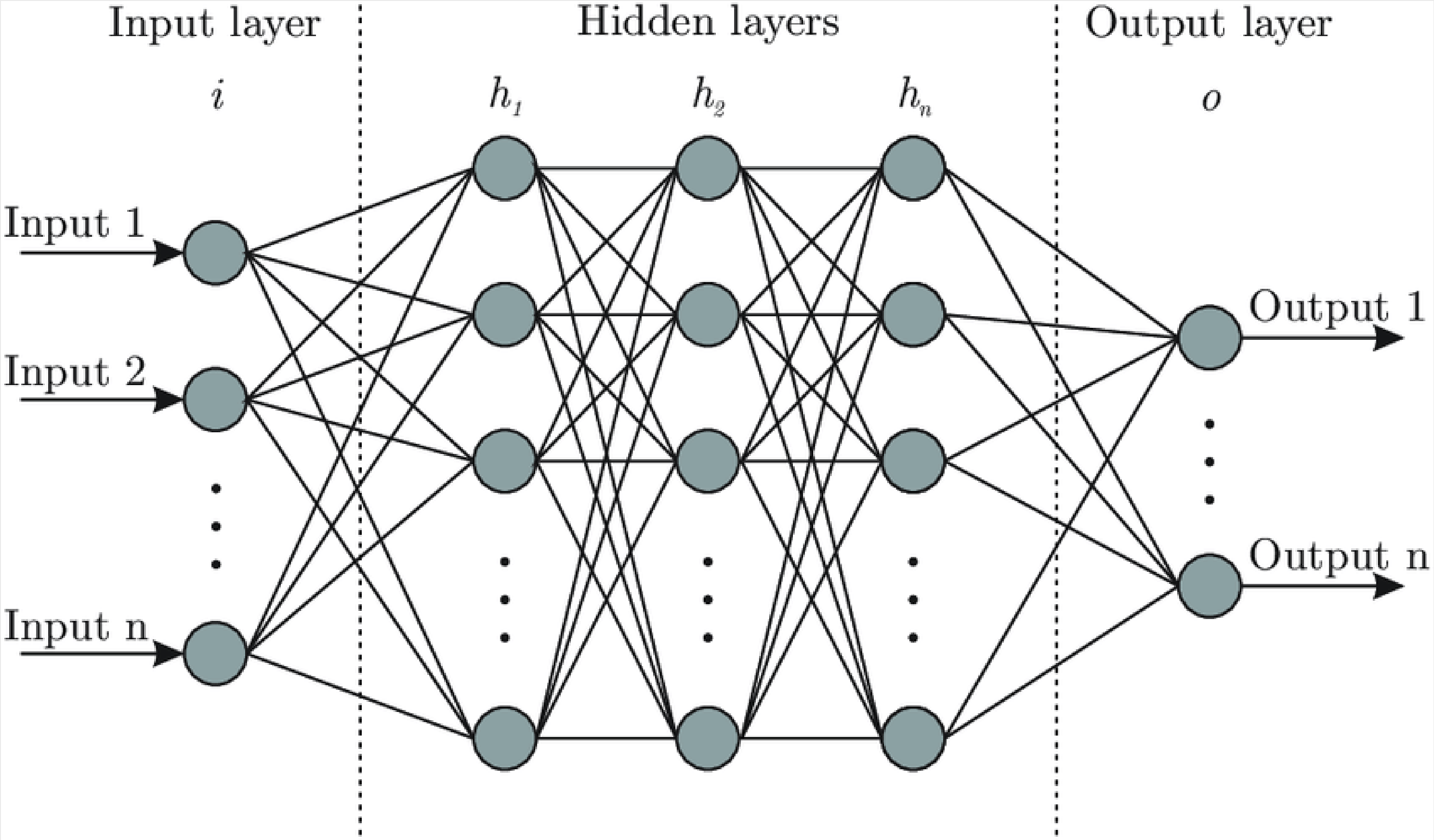
Activation Function

- **Common activation function - sigmoid**



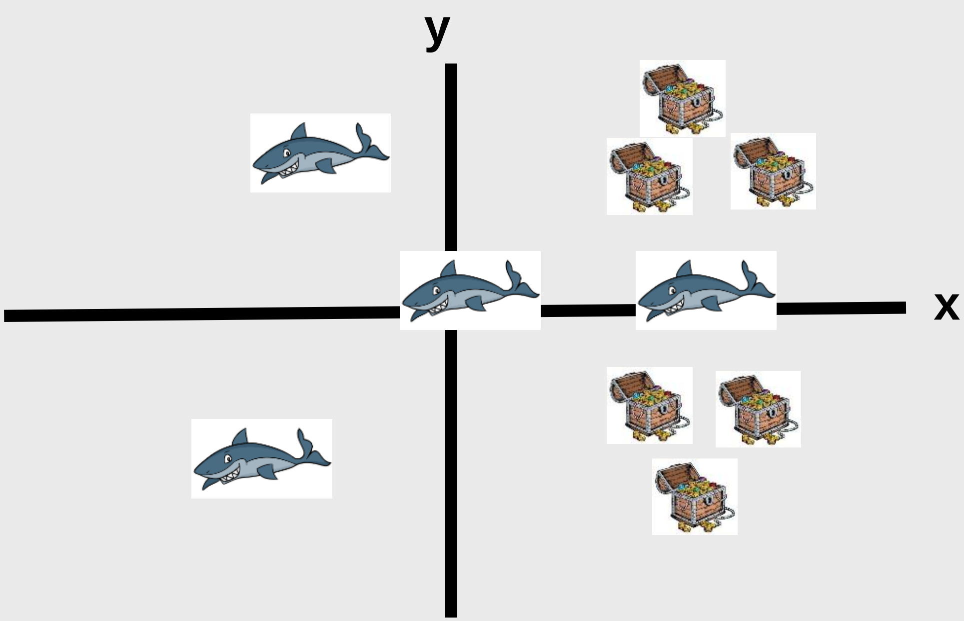
Deep Neural Network

- **We can have multiple layers of neurons**



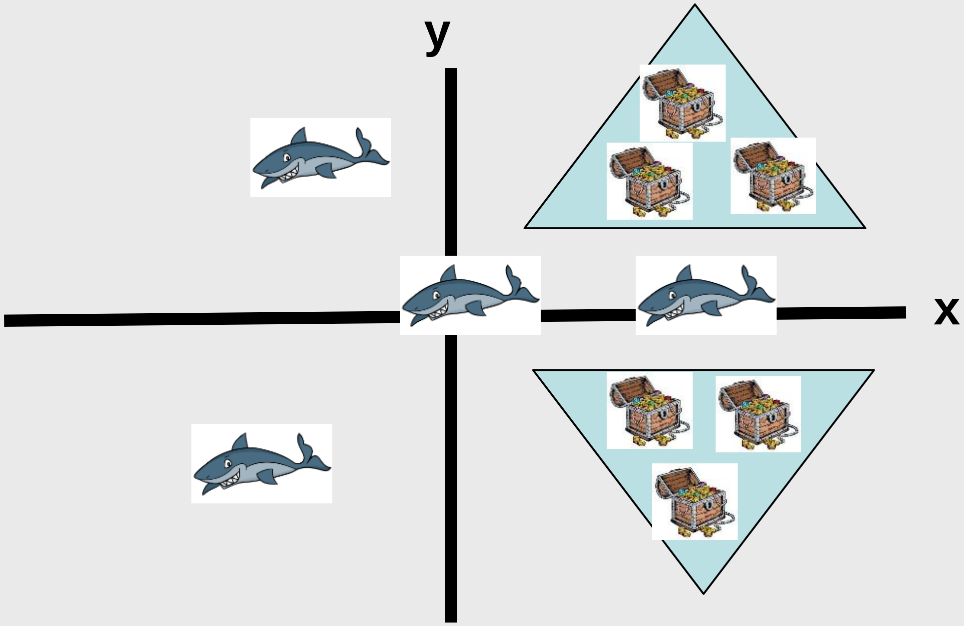
Classification Problem

- **Build a neural network to classify these points**



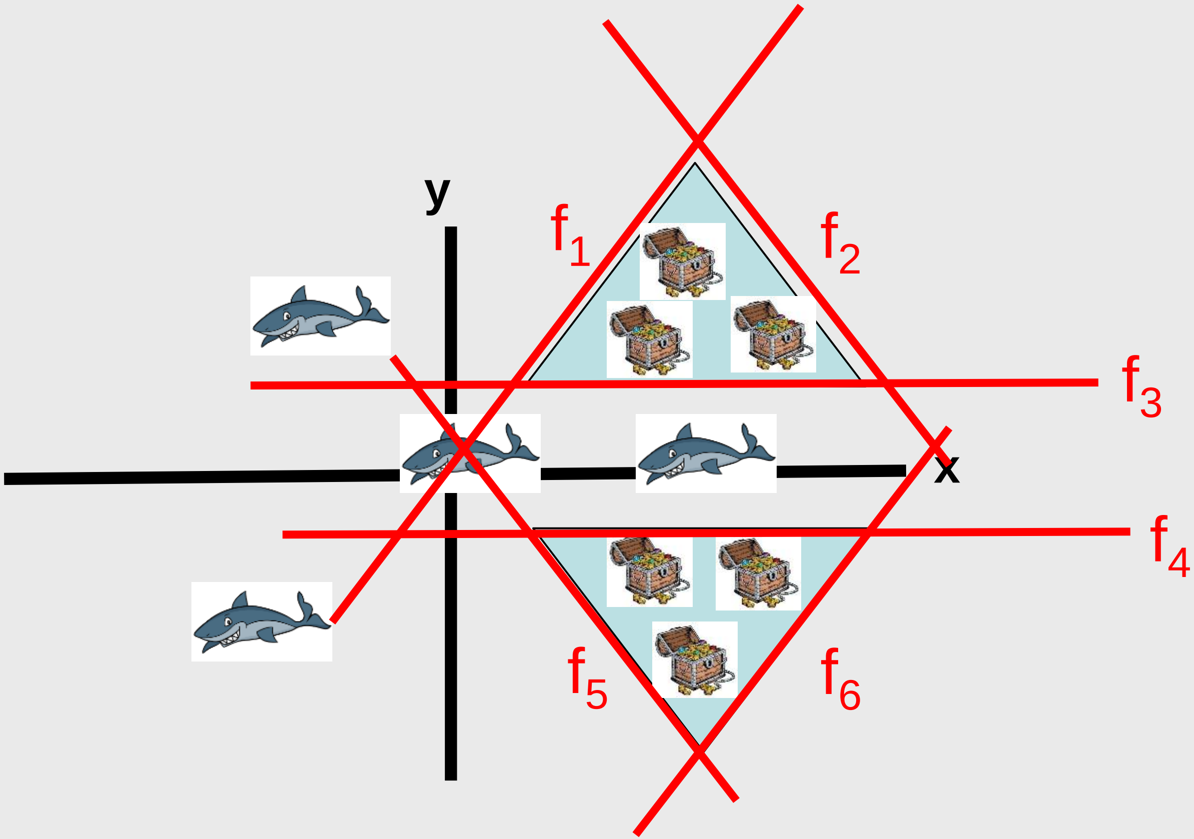
Classification Problem

- 2 classification regions – the 2 triangles



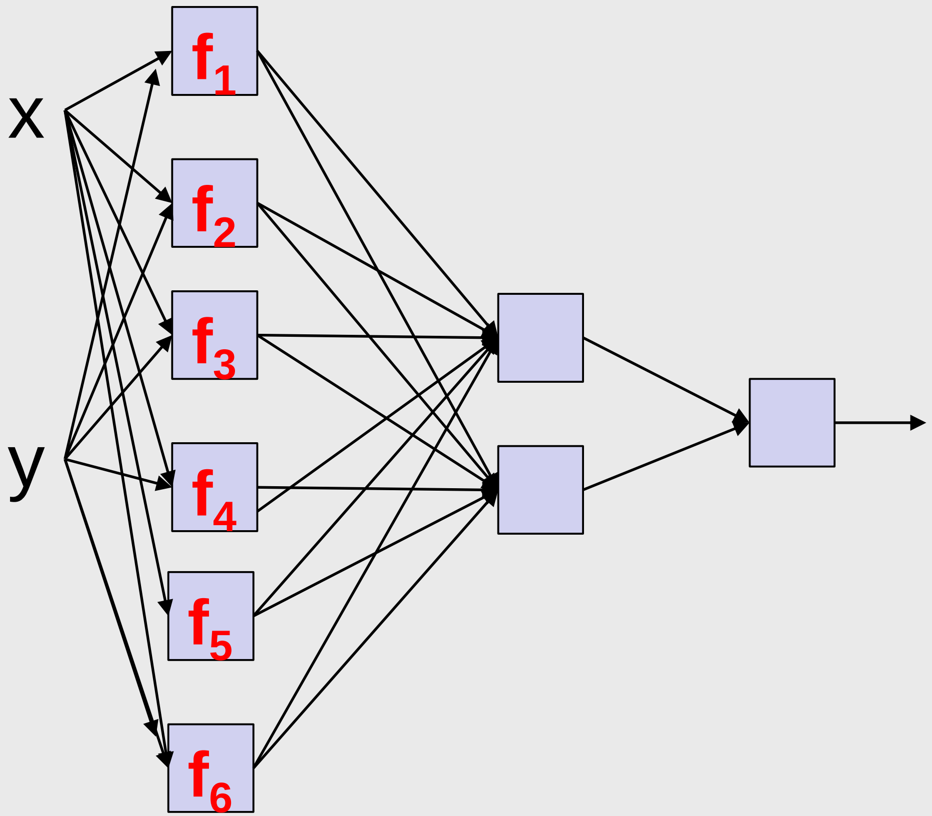
Neural Network Solution

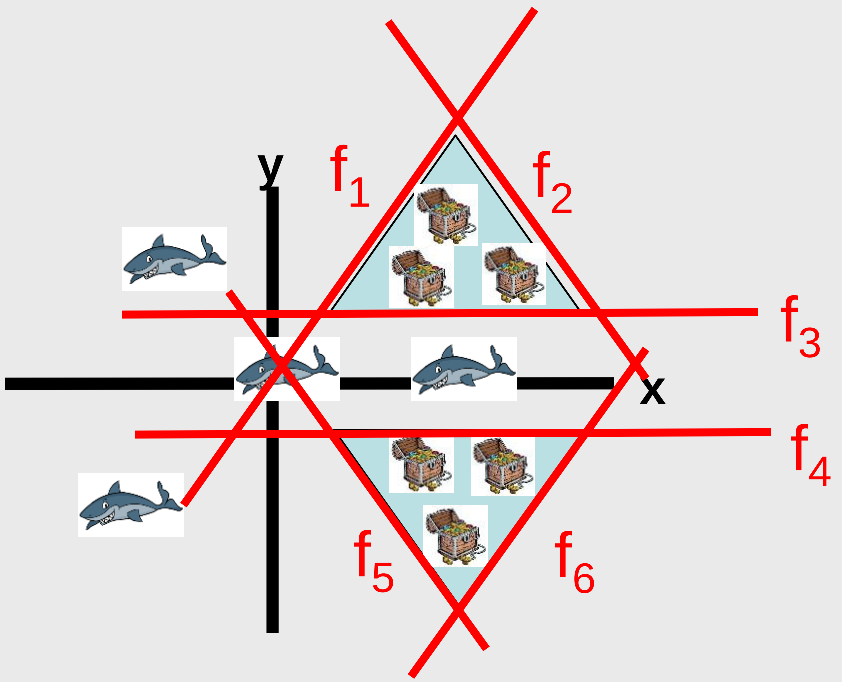
- 2 classification regions – the 2 triangles
- 6 features – one for each side of the triangles



Neural Network Solution

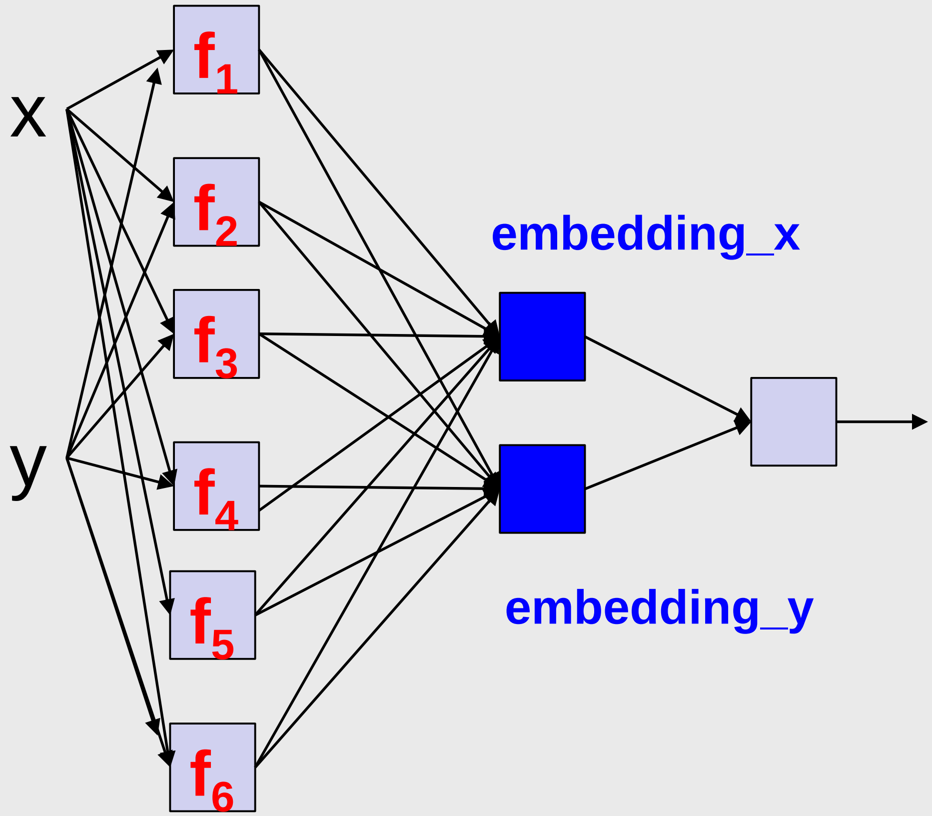
- Classify using a neural network



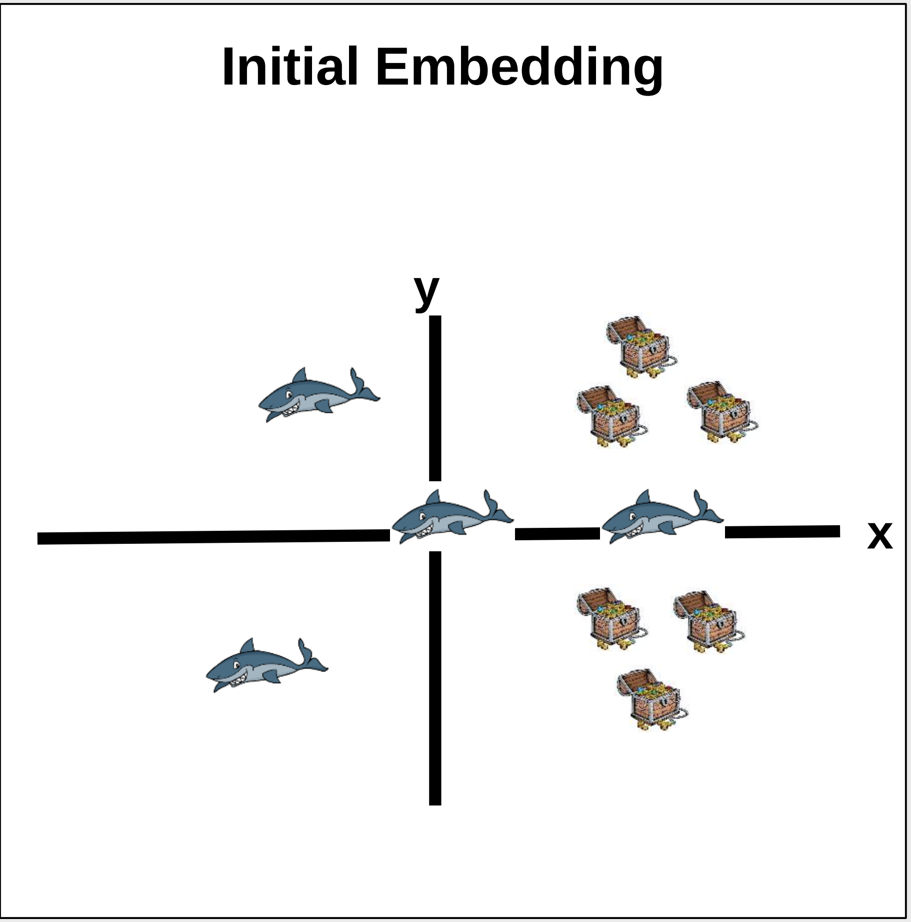


Neural Network Embedding

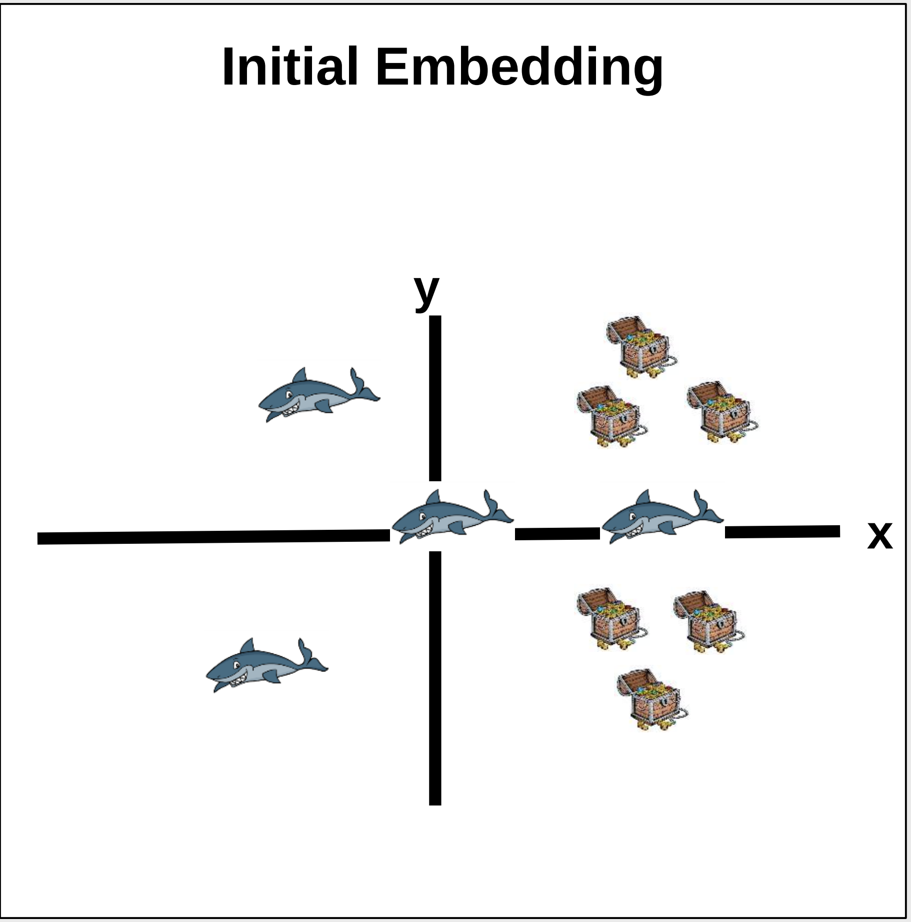
- Hidden layer can act as an embedding

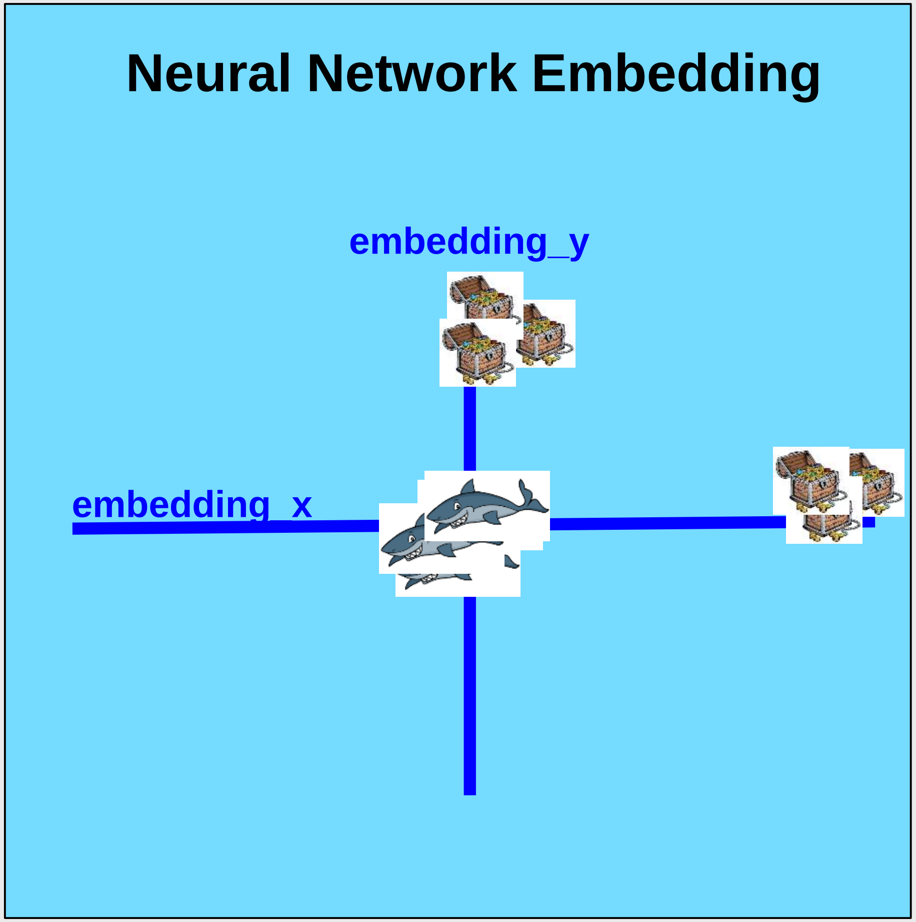


Neural Network Embedding



Neural Network Embedding

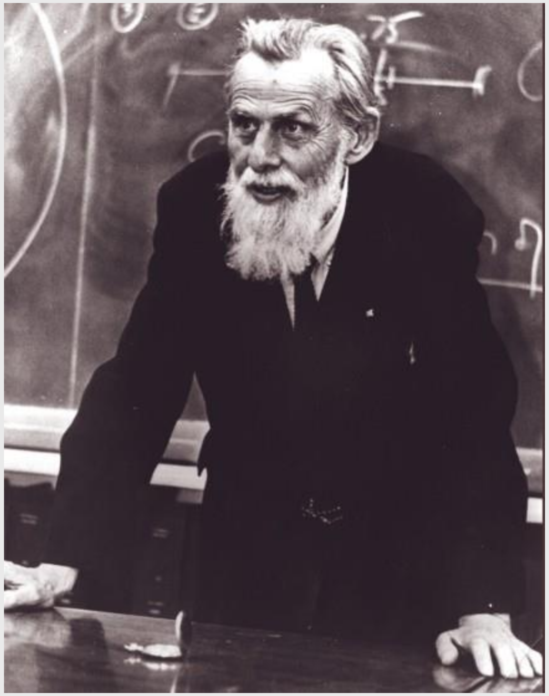




NEURAL NETWORK HISTORY

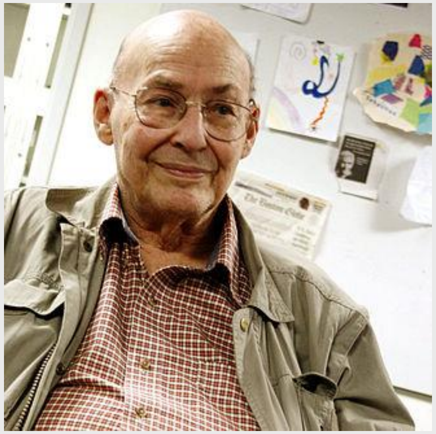
Origins of Neural Networks

- **1943 – Walter Pitts and Warren McCullough propose "nervous nets"**



Haters

- **1969 – Marvin Minsky and Seymour Papert say single layer neural network cant do that much AND computers are too slow to use neural nets**

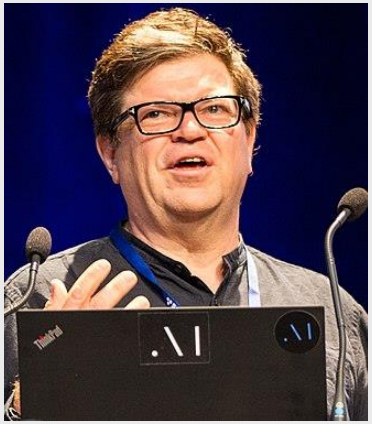


Haters

Neural Nets in the 1980s

- **1986 – Geoffrey Hinton and co-authors use back-propagation to train a neural network**

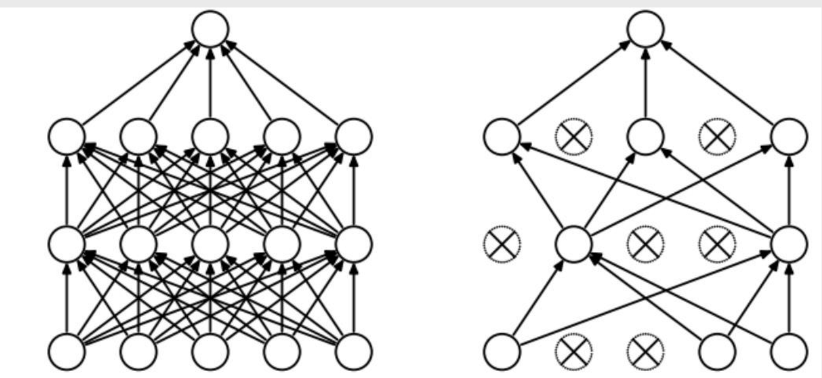
- **1989 – Yann LeCun creates convolutional neural networks to read zipcodes**



Neural Nets in the 2000s

The Beginning of Deep Learning

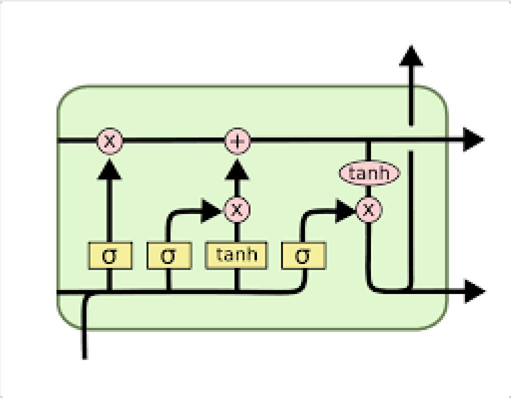
- **2012 – Hinton and his students win a drug discovery contest held by Merck – using a deep neural network trained using the “dropout” technique he invented**



(a) Standard Neural Net (b) After applying dropout.

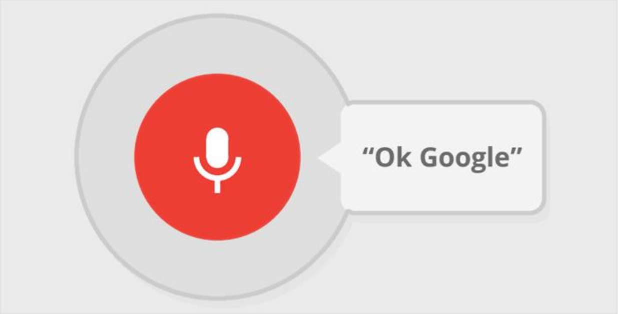
Long-term Short-Term Recurrent Neural Networks

- 1987 – Long-term short-term recurrent neural networks (LTSM RNN) invented



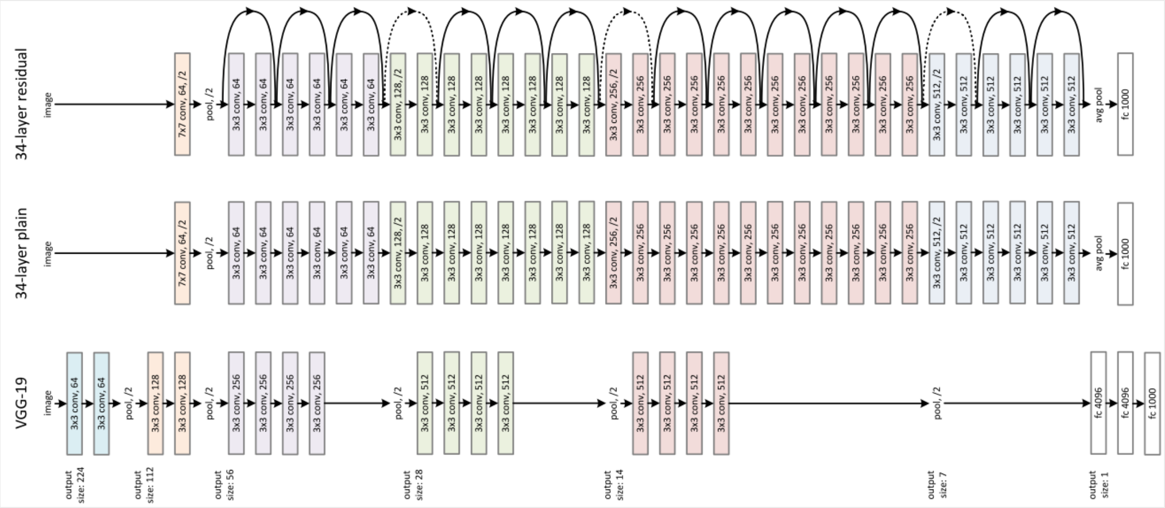
- 2010s – breakthroughs in speech recognition achieved using LTSM RNNs

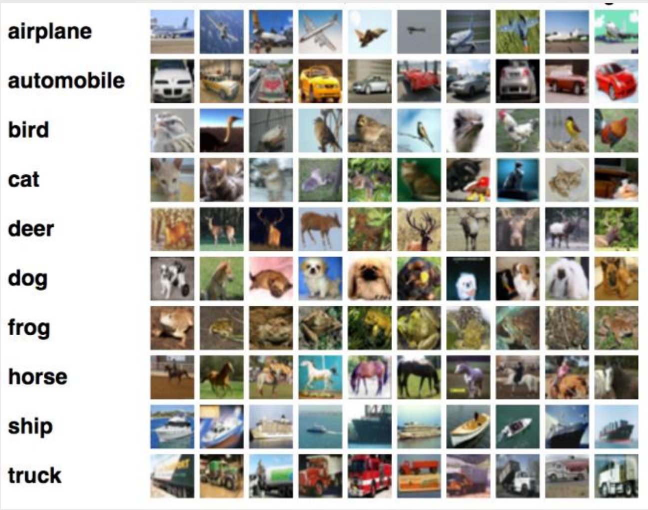
- Google voice search uses LTSM RNNs



Residual Nets

- 2015 – Residual Networks (ResNets) proposed
- Revolutionizes object recognition
- Error rates near 5%





Transformers

- Developed in 2017 by Google
- Revolutionized natural language processing

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention

What Can Transformers Do?

- Measure sentiment
- Translation
- Web search
- Text summarization
- Question answering
- Generate text

Transformer Architecture

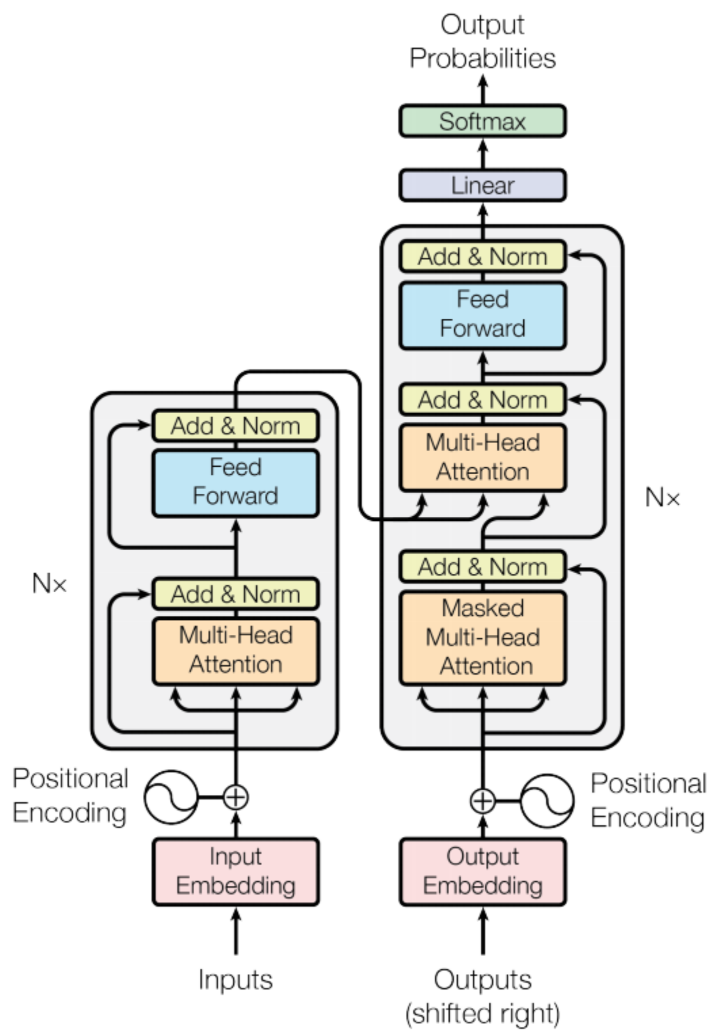
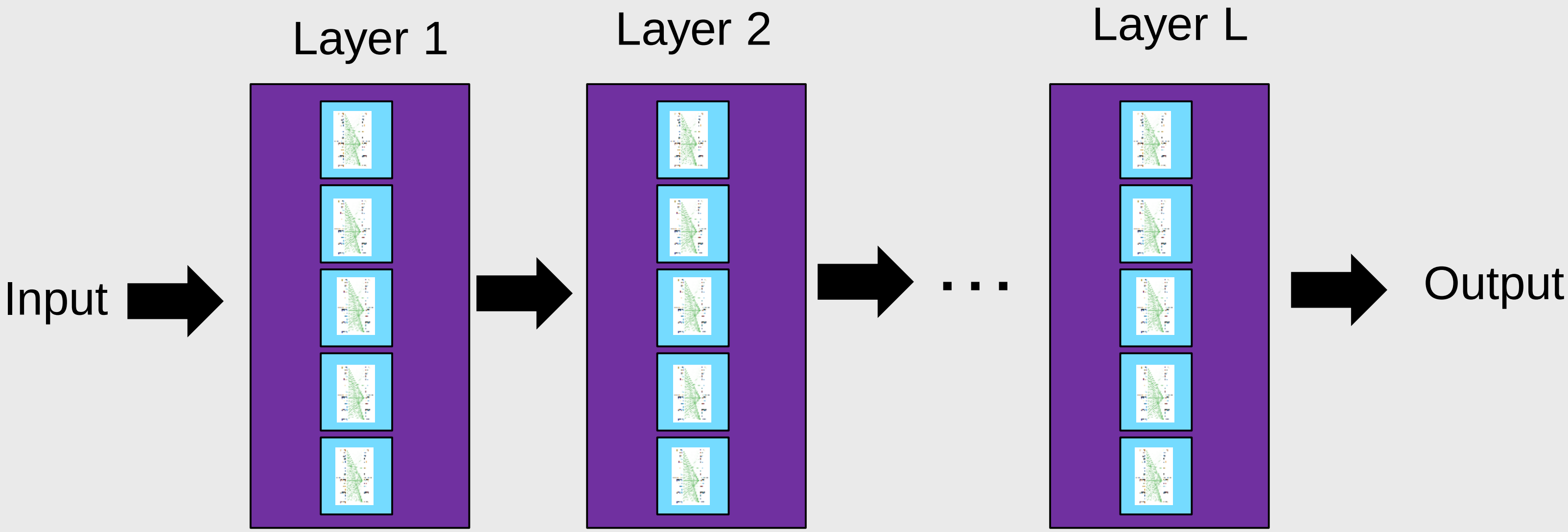


Figure 1: The Transformer - model architecture.

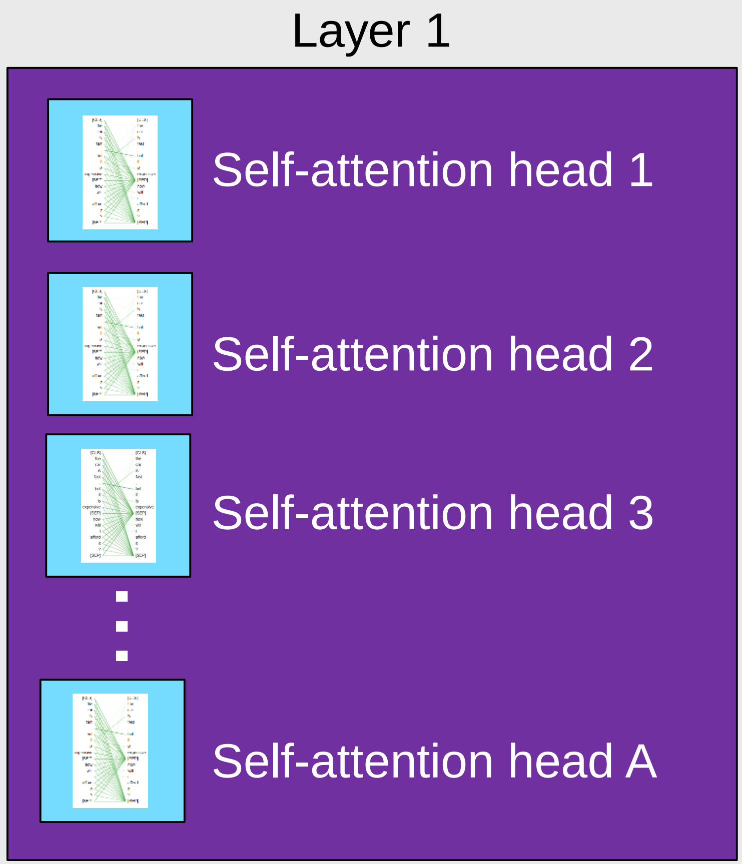
Transformer Architecture

- **The transformer has many layers**



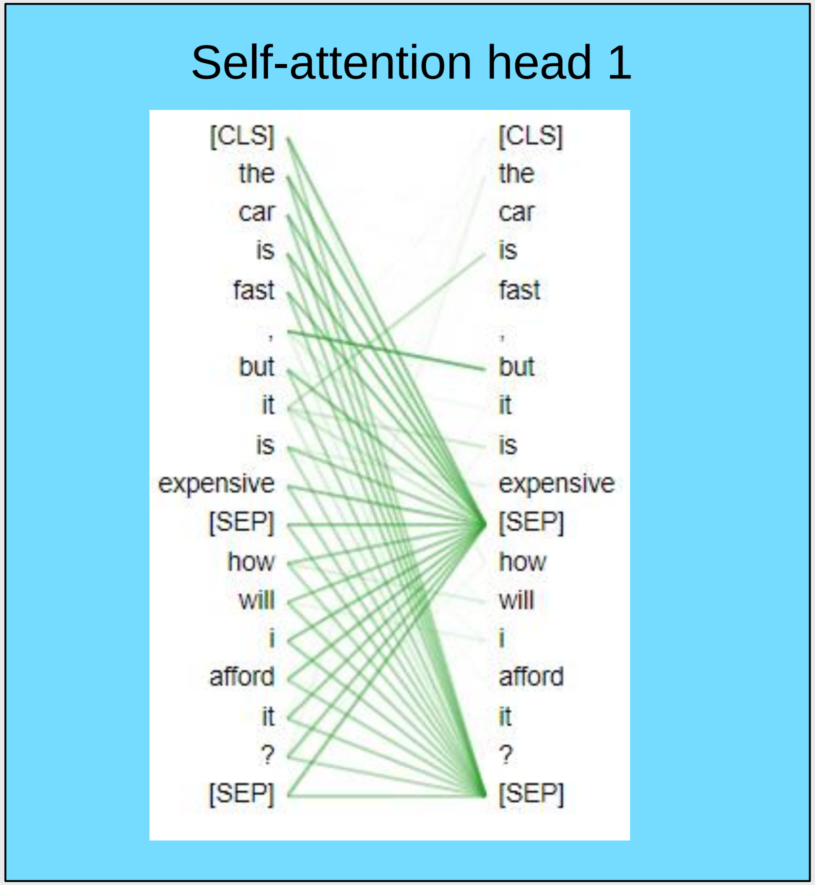
Transformer Layers

- **Each layer has many self-attention heads**

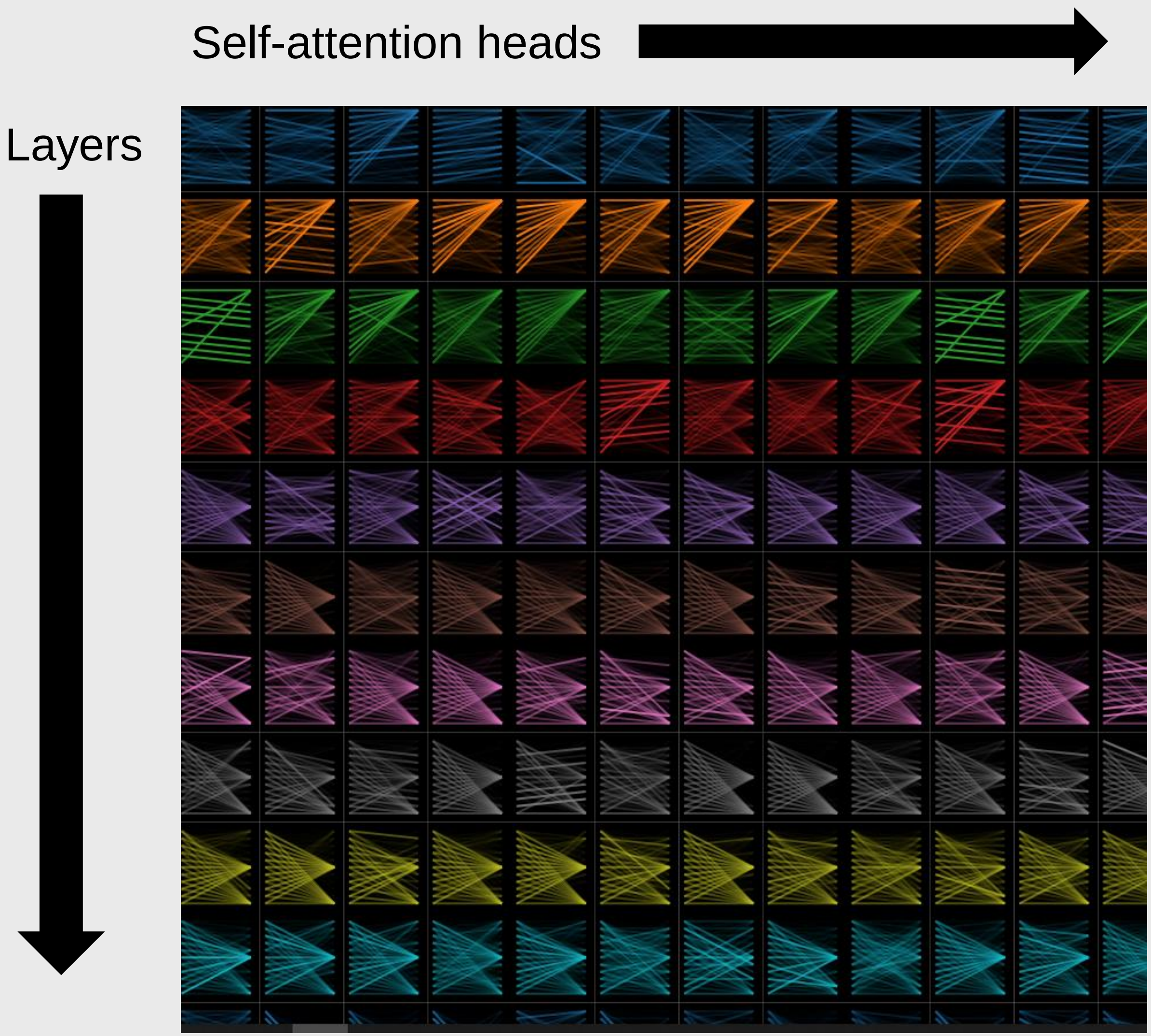


Self-Attention Head

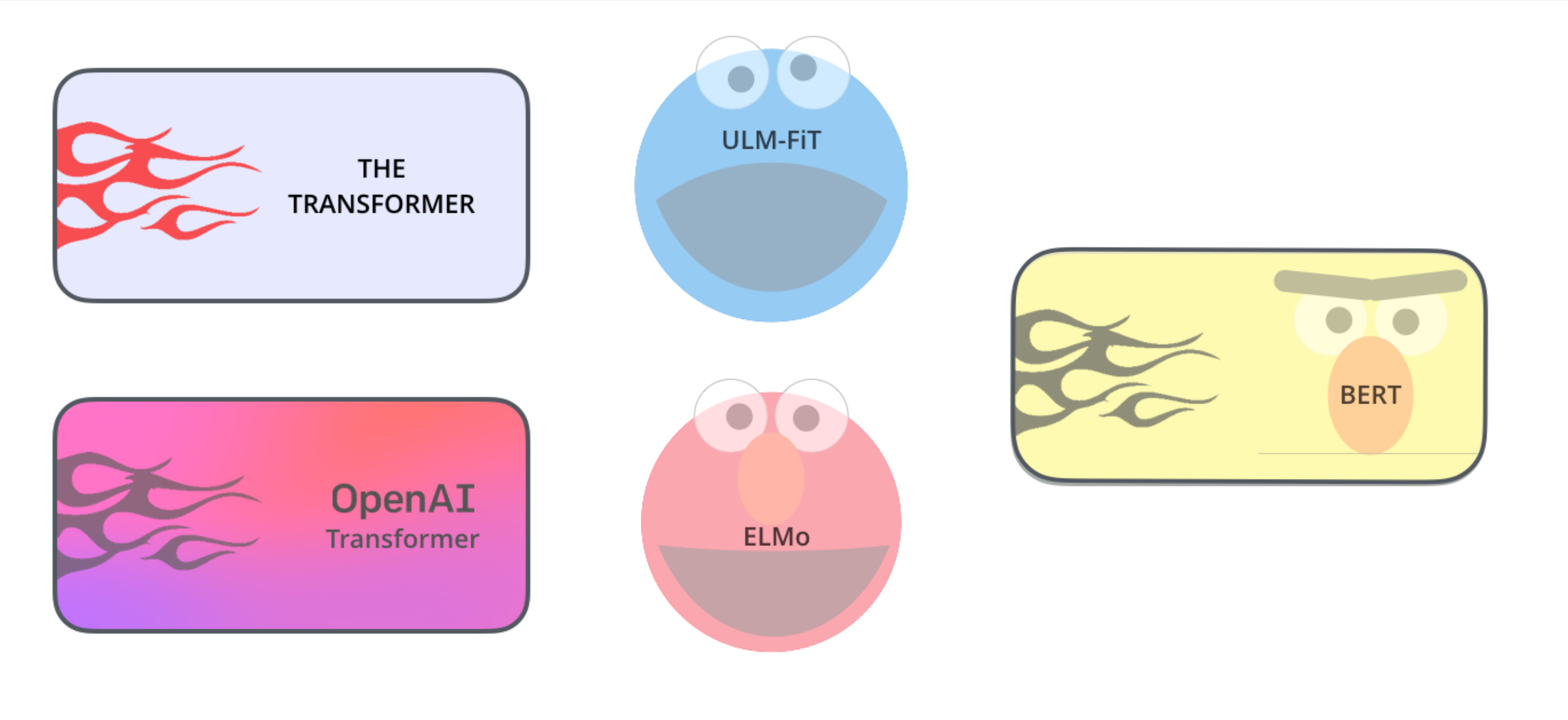
- **Each self-attention head contains attention weights from each word to each other word**



Visualizing the Brain of a Transformer

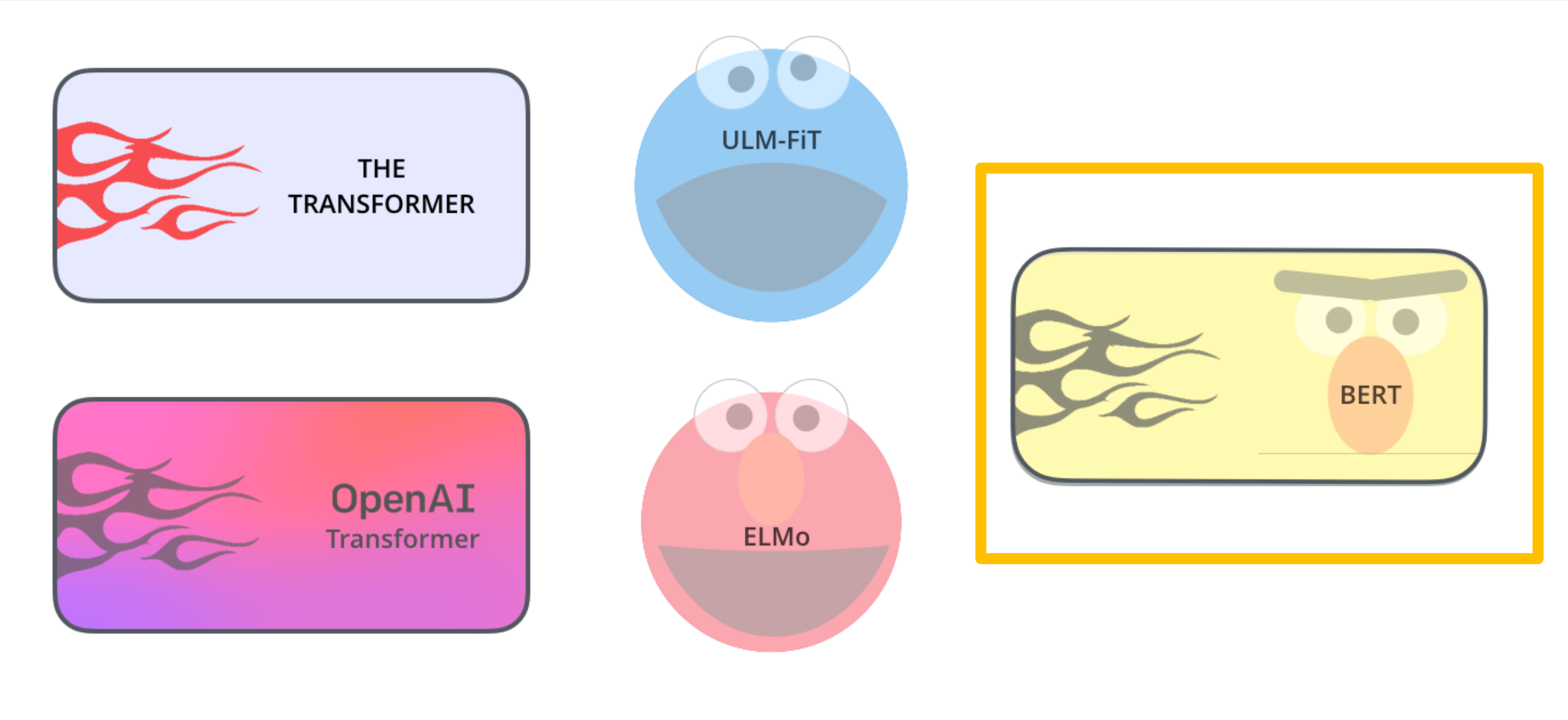


Popular Transformers



http://jalammar.github.io/illustrated-bert/

Popular Transformers

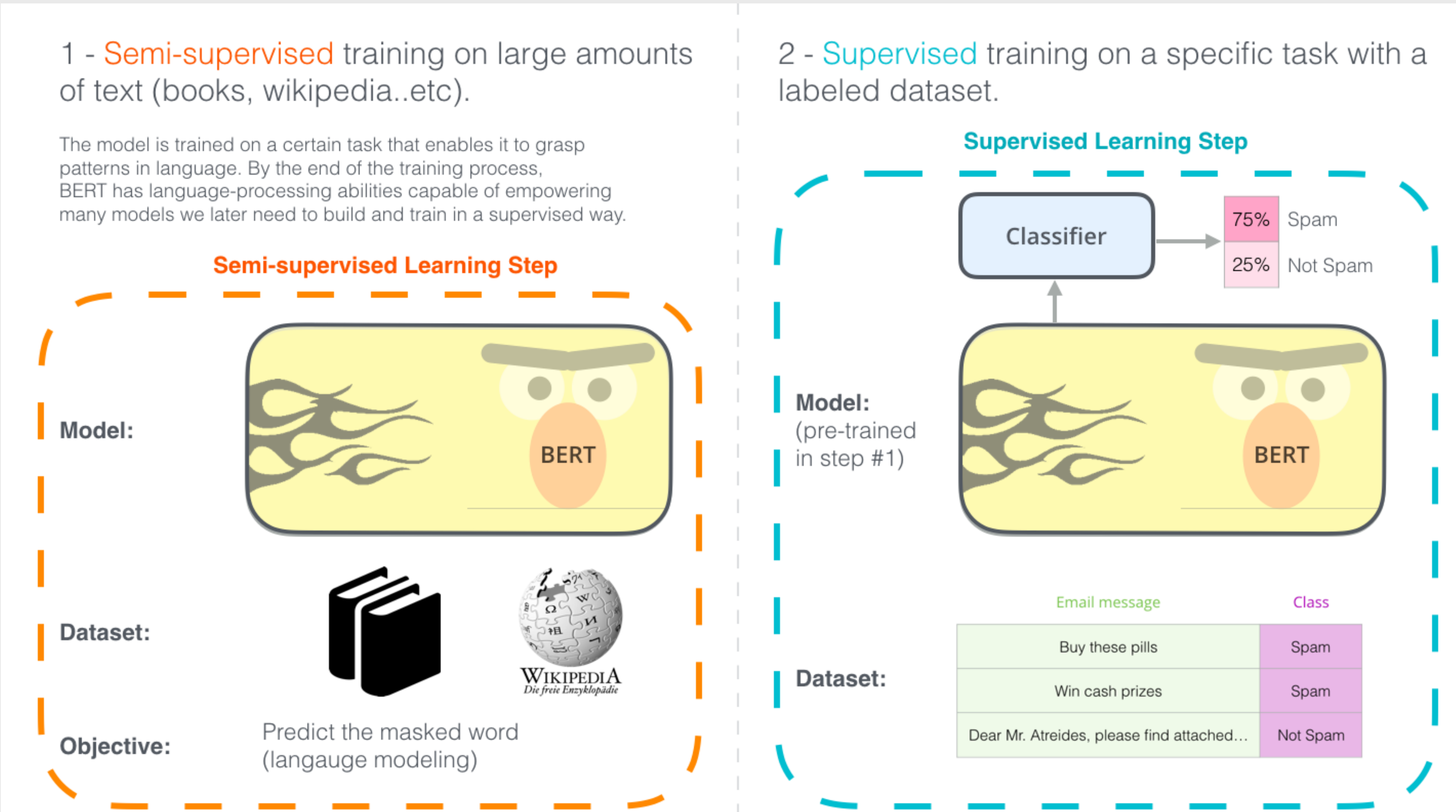


http://jalammar.github.io/illustrated-bert/

BERT

- BERT = Bi-directional Encoder Representations From Transformers
- Released in 2018 by Google
- Base BERT has 100 million parameters
  - 12 layers
  - 12 attention heads
  - 768 dimensional word embedding
- Trained on books and Wikipedia (3.3 billion words)

Training BERT



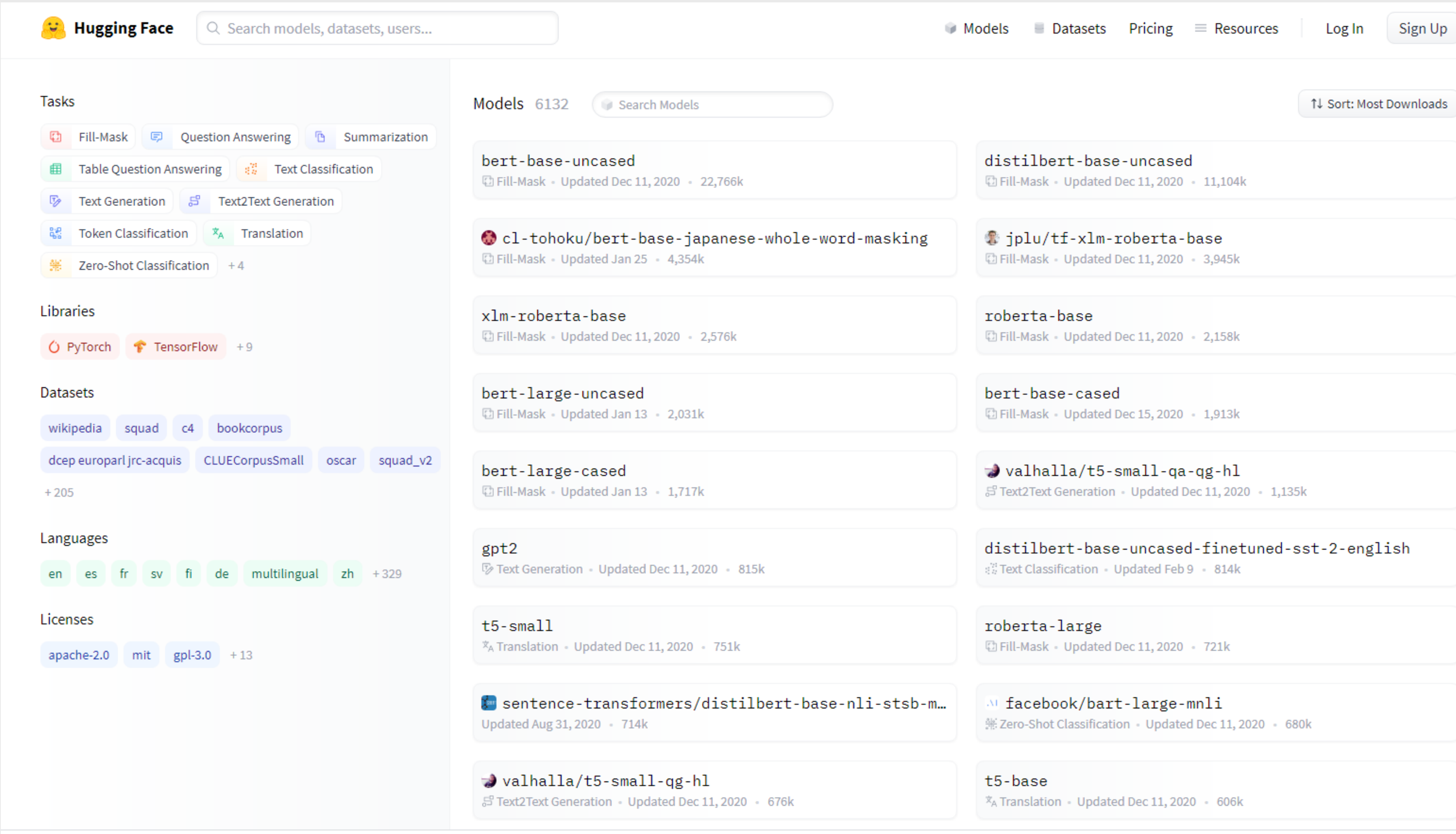
http://jalammar.github.io/illustrated-bert/

Masked Language Model Task

- **BERT is trained to learn a masked language model**
  - **Guess [MASK] words in a sentence**

Data	Prediction
I went to the [MASK] to buy milk.	[MASK] = store
I graduated from [MASK] and got a degree.	[MASK] = college
I had a [MASK] and it tasted [MASK]!	[MASK] = hamburger [MASK] = amazing

Pre-Trained Transformers: Hugging Face



Evaluating Language Models: GLUE

- GLUE = general language understanding and evaluation
- GLUE is a set of benchmark tasks to evaluate language models like BERT

GLUE Tasks

Task type	Description
Acceptability	Is the sentence grammatically correct
Sentiment	Can you predict the sentiment of the sentence
Question answering	Does the second sentence answer the question in the first sentence
Natural language inference	Does the second sentence entail the hypothesis in the first sentence
Pronoun referral	To what does the pronoun in a sentence refer
Sentence similarity	Are the two sentences paraphrases of each other

GLUE Leaders

- **Human GLUE score = 87.1**
- **GLUE leaderboard: https://gluebenchmark.com/leaderboard**

Rank	Name	Model	URL	Score
1	ERNIE Team - Baidu	ERNIE		90.9
2	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.8
3	HFL iFLYTEK	MacALBERT + DKM		90.7
4	Alibaba DAMO NLP	StructBERT + TAPT		90.6
5	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS		90.6
6	T5 Team - Google	T5		90.3
7	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART		89.9
8	Huawei Noah's Ark Lab	NEZHA-Large		89.8
9	Zihang Dai	Funnel-Transformer (Ensemble B10-10-10H1024)		89.7
10	ELECTRA Team	ELECTRA-Large + Standard Tricks		89.4

SuperGLUE Leaders

- Human SuperGLUE score = 89.8
- SuperGLUE leaderboard: https://super.gluebenchmark.com/leaderboard

	Rank	Name	Model	URL	Score
+	1	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
+	2	Zirui Wang	T5 + Meena, Single Model (Meena Team - Google Brain)		90.2
	3	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8
+	4	T5 Team - Google	T5		89.3
+	5	Huawei Noah's Ark Lab	NEZHA-Plus		86.7
+	6	Alibaba PAI&ICBU	PAI Albert		86.1
+	7	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9
+	8	Infosys : DAWN : AI Research	RoBERTa-iCETS		85.8
	9	Zhuiyi Technology	RoBERTa-mtl-adv		85.7
	10	Facebook AI	RoBERTa		84.6

SuperGLUE Leaders

- **Human SuperGLUE score = 89.8**
- **SuperGLUE leaderboard: https://super.gluebenchmark.com/leaderboard**

Rank	Name	Model	URL	Score
1	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4		90.3
2	Zirui Wang	T5 + Meena, Single Model (Meena Team - Google Brain)		90.2
3	SuperGLUE Human Baselines	SuperGLUE Human Baselines		89.8
4	T5 Team - Google	T5		89.3
5	Huawei Noah's Ark Lab	NEZHA-Plus		86.7
6	Alibaba PAI&ICBU	PAI Albert		86.1
7	Tencent Jarvis Lab	RoBERTa (ensemble)		85.9
8	Infosys : DAWN : AI Research	RoBERTa-iCETS		85.8
9	Zhuiyi Technology	RoBERTa-mtl-adv		85.7
10	Facebook AI	RoBERTa		84.6

Next Time: Coding Session

- **Use a pre-trained BERT sentiment classifier to measure tweet sentiment**

- **Learn how to use any model in the huggingface library**

- **Need to install some neural network packages – please try this before class because there might be issues**