

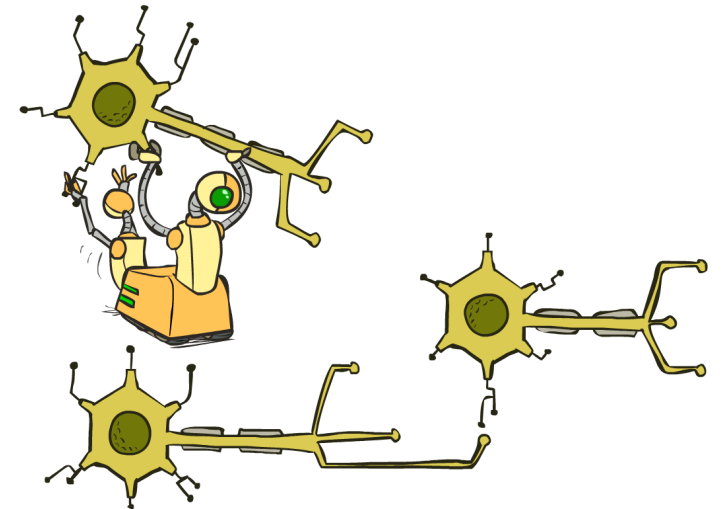
---

# CSE 573 :

# Artificial Intelligence

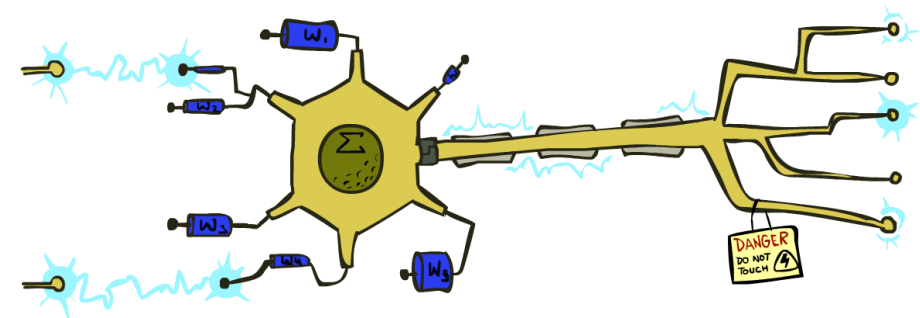
Hanna Hajishirzi  
Neural Networks

slides adapted from  
Dan Klein, Pieter Abbeel [ai.berkeley.edu](http://ai.berkeley.edu)  
And Dan Weld, Luke Zettlemoyer



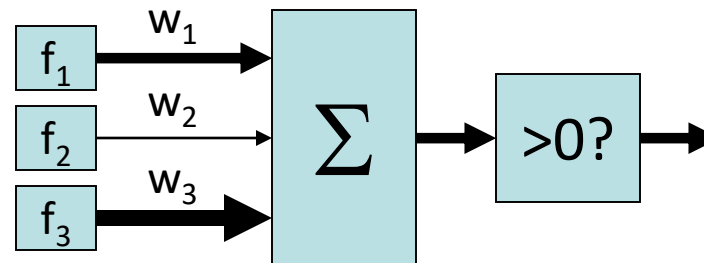
# Reminder: Linear Classifiers

- Inputs are **feature values**
- Each feature has a **weight**
- Sum is the **activation**



$$\text{activation}_w(x) = \sum_i w_i \cdot f_i(x) = w \cdot f(x)$$

- If the activation is:
  - Positive, output +1
  - Negative, output -1

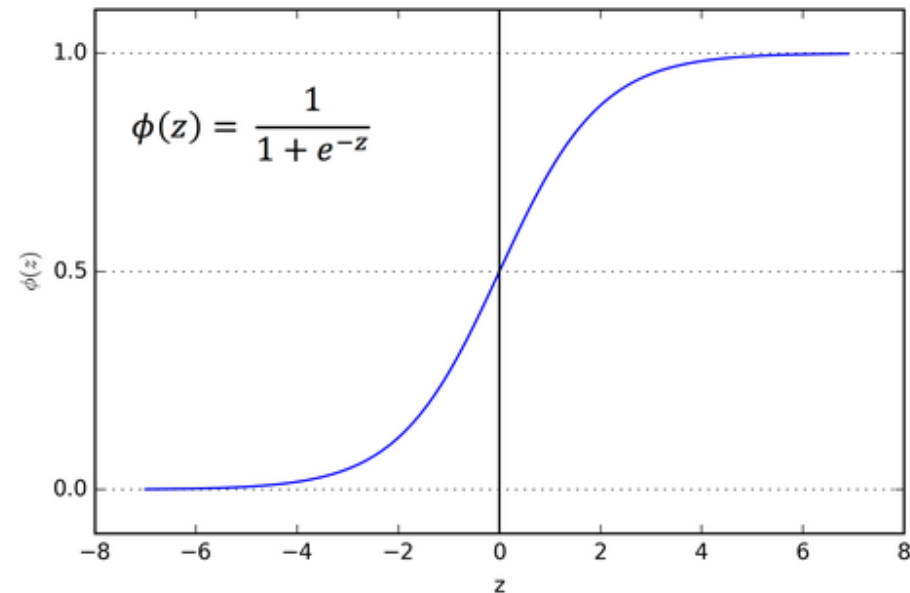


# How to get probabilistic decisions?

- Activation:  $z = w \cdot f(x)$
- If  $z = w \cdot f(x)$  very positive  $\rightarrow$  want probability going to 1
- If  $z = w \cdot f(x)$  very negative  $\rightarrow$  want probability going to 0

- Sigmoid function

$$\phi(z) = \frac{1}{1 + e^{-z}}$$



# Best $w$ ?

- Maximum likelihood estimation:

$$\max_w ll(w) = \max_w \sum_i \log P(y^{(i)} | x^{(i)}; w)$$

with:  $P(y^{(i)} = +1 | x^{(i)}; w) = \frac{1}{1 + e^{-w \cdot f(x^{(i)})}}$

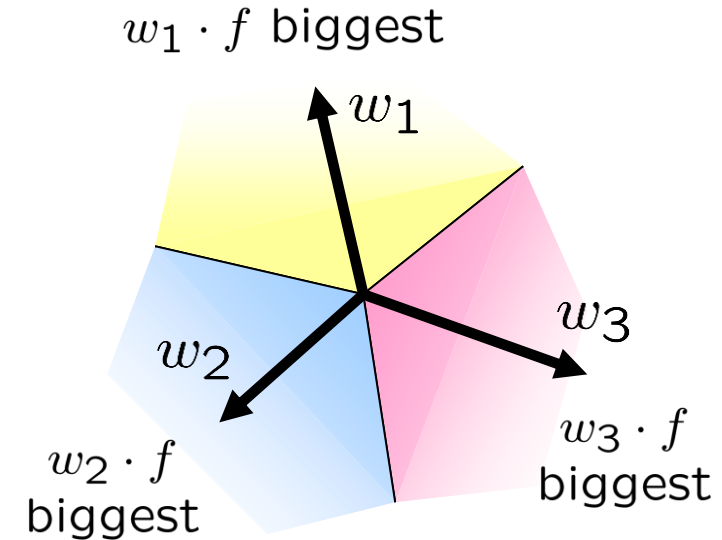
$$P(y^{(i)} = -1 | x^{(i)}; w) = 1 - \frac{1}{1 + e^{-w \cdot f(x^{(i)})}}$$

**= Logistic Regression**

# Multiclass Logistic Regression

- Multi-class linear classification

- A weight vector for each class:  $w_y$
- Score (activation) of a class  $y$ :  $w_y \cdot f(x)$
- Prediction w/highest score wins:  $y = \arg \max_y w_y \cdot f(x)$



- How to make the scores into probabilities?

$$\underbrace{z_1, z_2, z_3}_{\text{original activations}} \rightarrow \underbrace{\frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}, \frac{e^{z_2}}{e^{z_1} + e^{z_2} + e^{z_3}}, \frac{e^{z_3}}{e^{z_1} + e^{z_2} + e^{z_3}}}_{\text{softmax activations}}$$

# Best $w$ ?

- Maximum likelihood estimation:

$$\max_w ll(w) = \max_w \sum_i \log P(y^{(i)} | x^{(i)}; w)$$

with:

$$P(y^{(i)} | x^{(i)}; w) = \frac{e^{w_{y^{(i)}} \cdot f(x^{(i)})}}{\sum_y e^{w_y \cdot f(x^{(i)})}}$$

**= Multi-Class Logistic Regression**

# This Lecture

---

- Optimization

- i.e., how do we solve:

$$\max_w ll(w) = \max_w \sum_i \log P(y^{(i)} | x^{(i)}; w)$$

# Hill Climbing

- simple, general idea
  - Start wherever
  - Repeat: move to the best neighboring state
  - If no neighbors better than current, quit
- What's particularly tricky when hill-climbing for multiclass logistic regression?
  - Optimization over a continuous space
    - Infinitely many neighbors!
    - How to do this efficiently?





# Mini-Batch Gradient Ascent on the Log Likelihood Objective

$$\max_w ll(w) = \max_w \sum_i \log P(y^{(i)} | x^{(i)}; w)$$

**Observation:** gradient over small set of training examples (=mini-batch) can be computed in parallel, might as well do that instead of a single one

- `init w`
- `for iter = 1, 2, ...`
  - pick random subset of training examples  $J$

$$w \leftarrow w + \alpha * \sum_{j \in J} \nabla \log P(y^{(j)} | x^{(j)}; w)$$

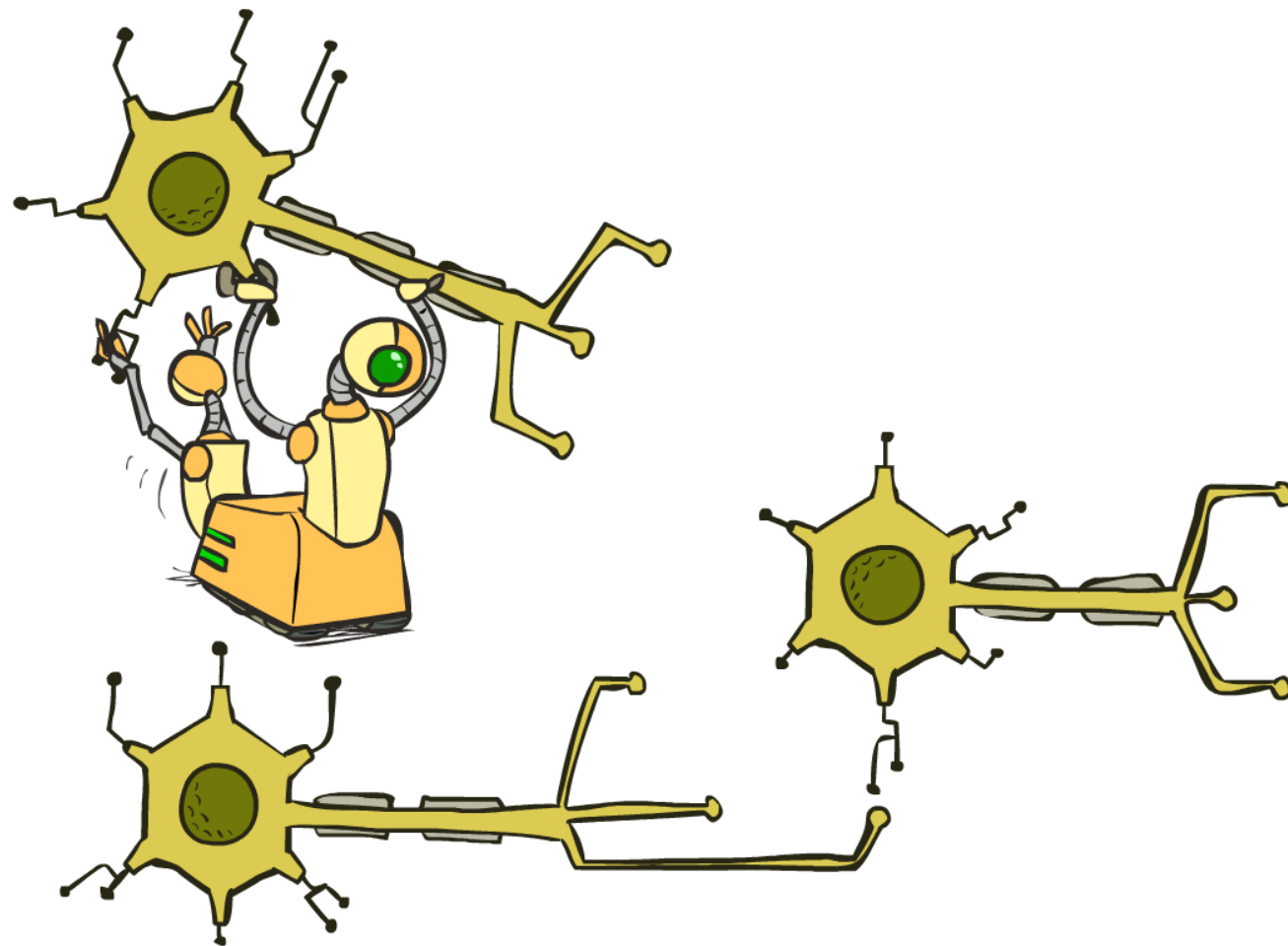
# How about computing all the derivatives?

---

- We'll talk about that once we covered neural networks, which are a generalization of logistic regression

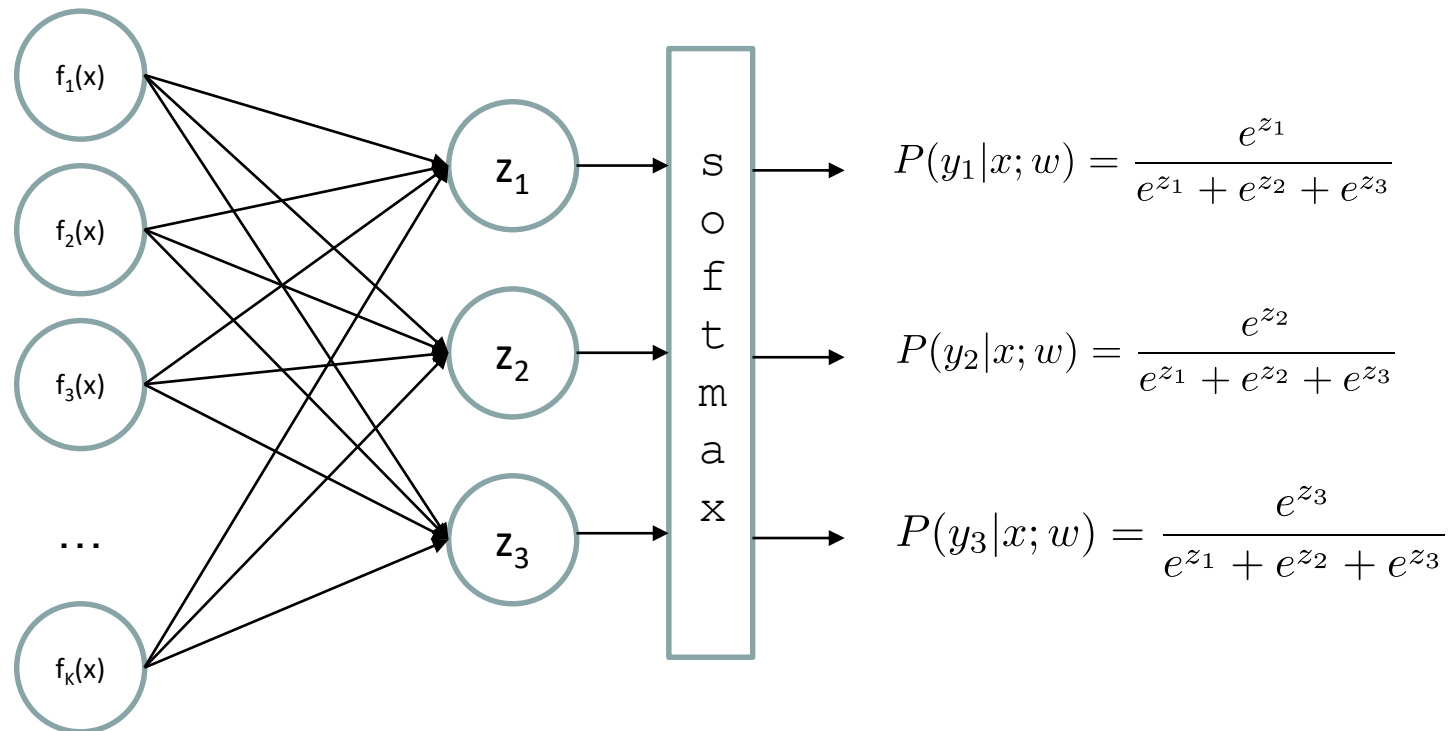
# Neural Networks

---

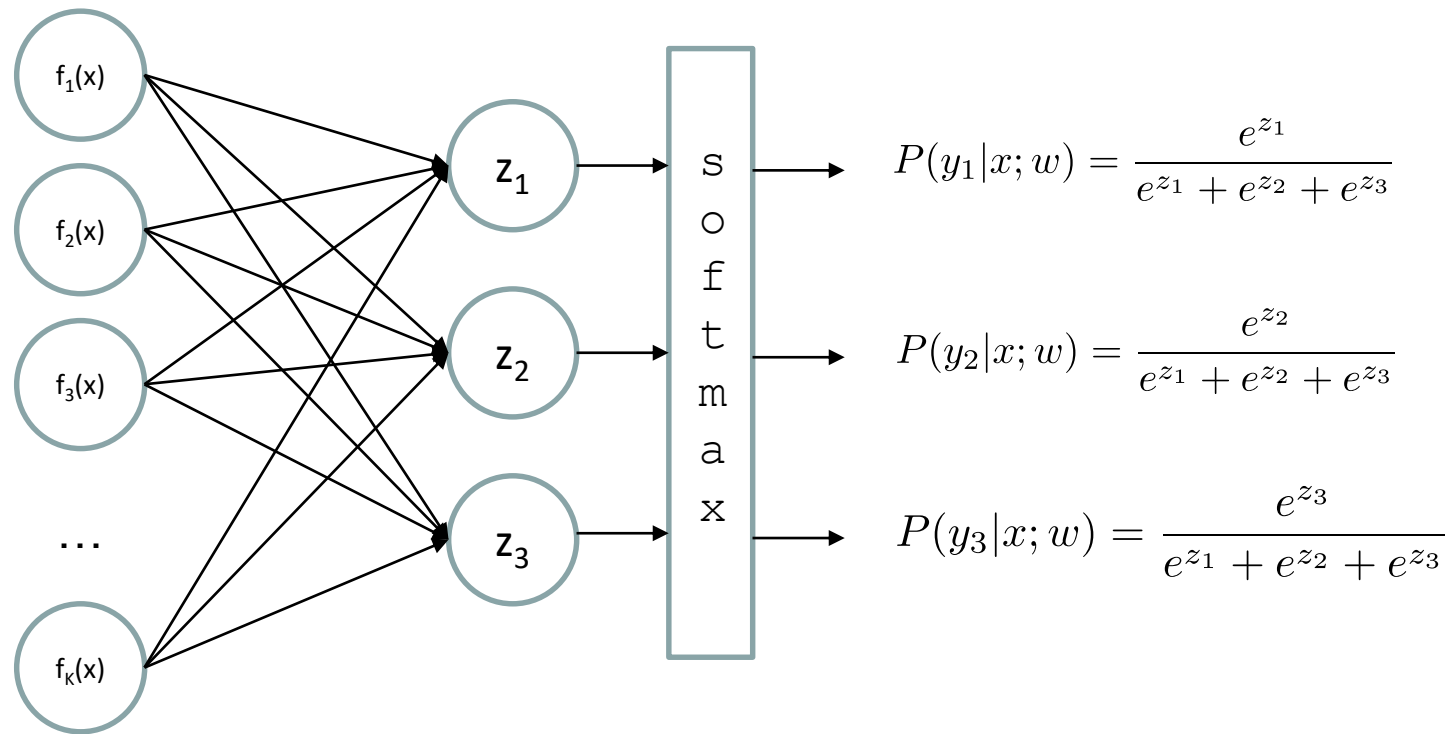


# Multi-class Logistic Regression

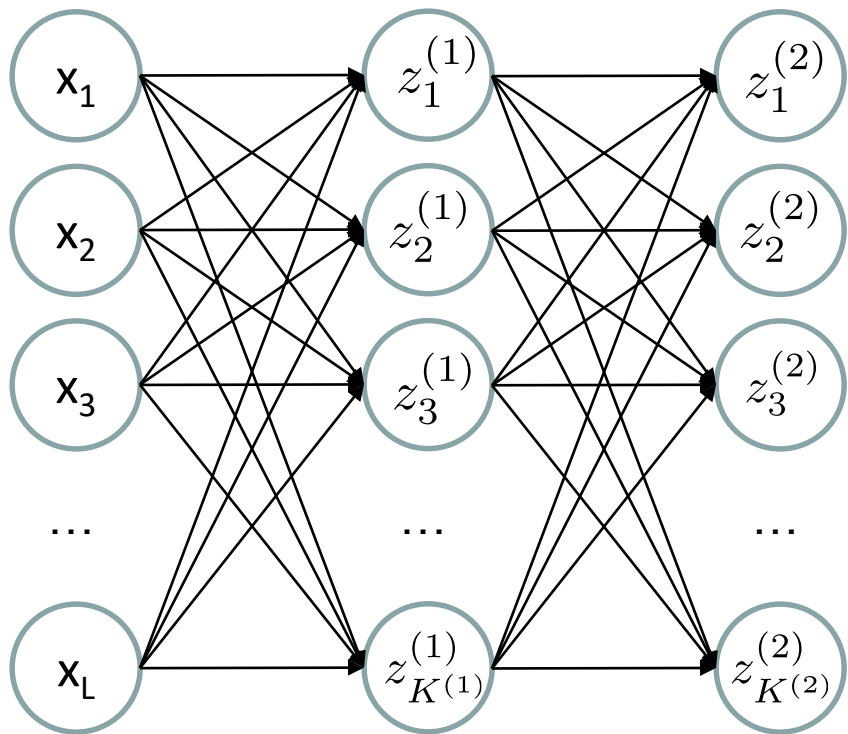
- = special case of neural network



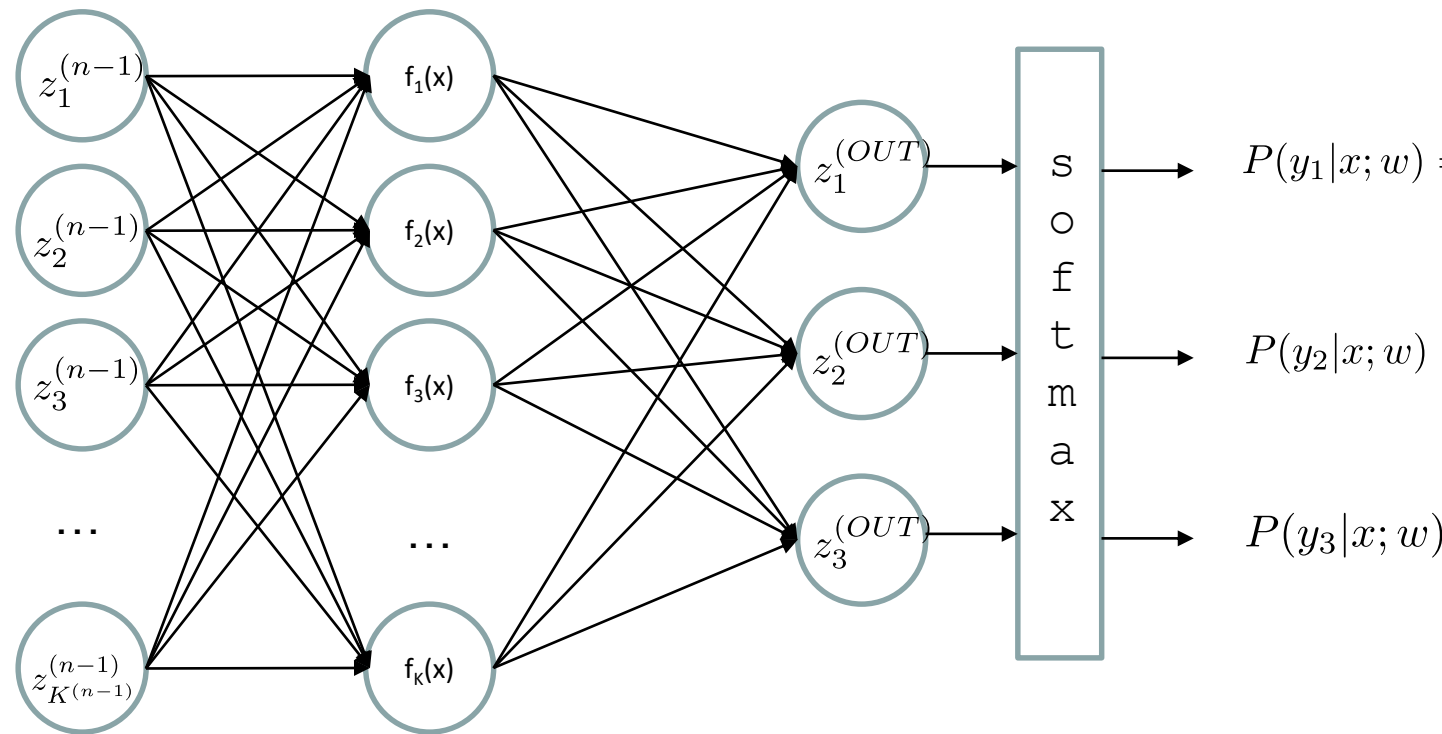
# Deep Neural Network = Also learn the features!



# Deep Neural Network = Also learn the features!



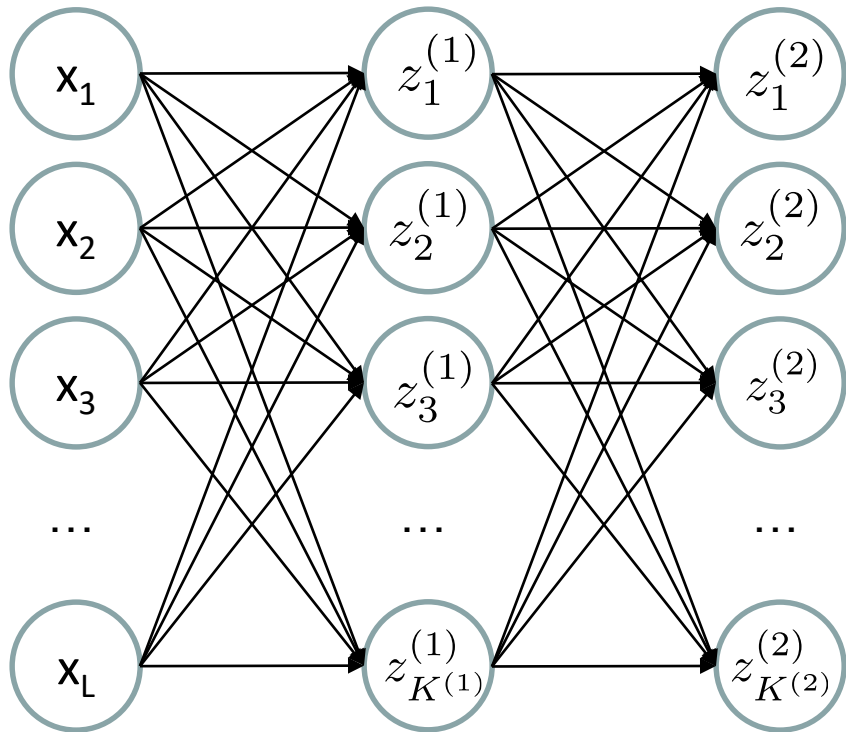
...



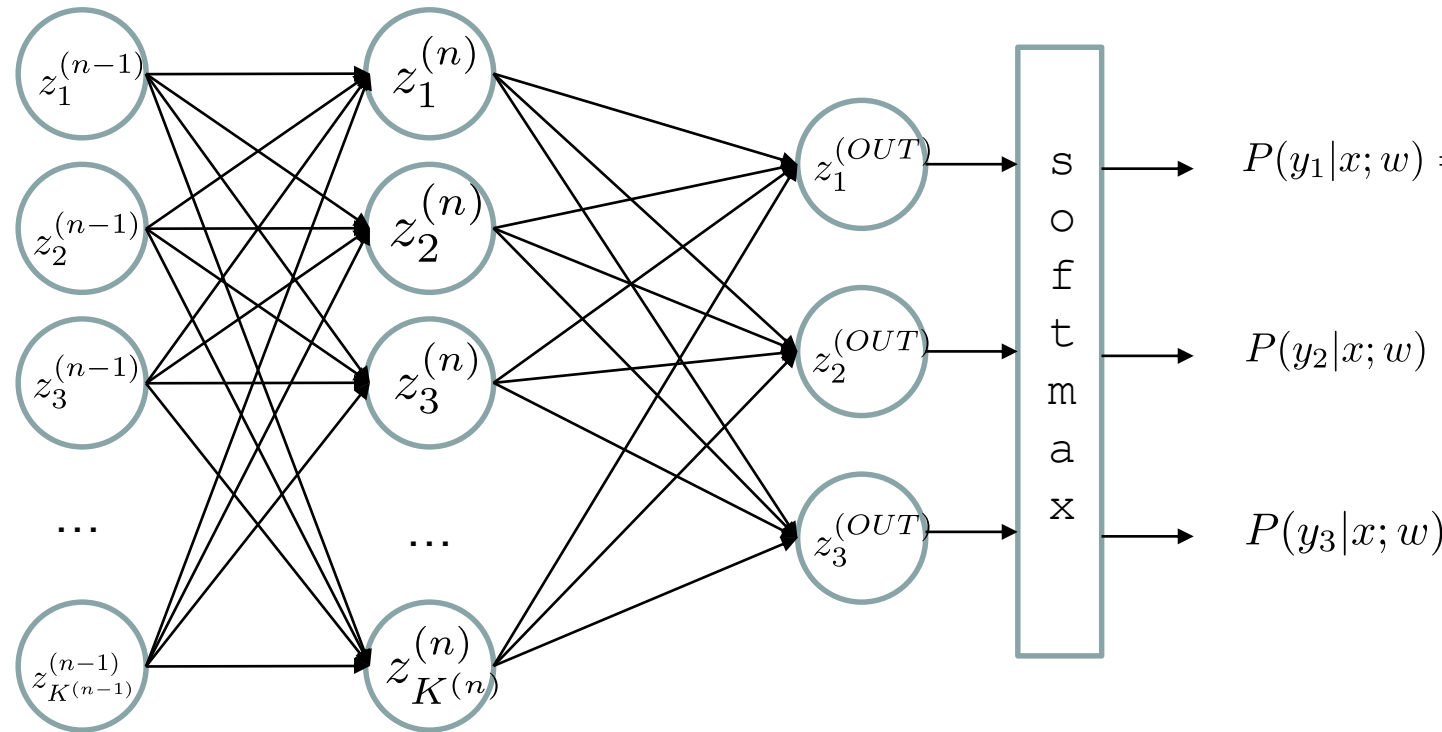
$$z_i^{(k)} = g\left(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)}\right)$$

**g = nonlinear activation function**

# Deep Neural Network = Also learn the features!



...

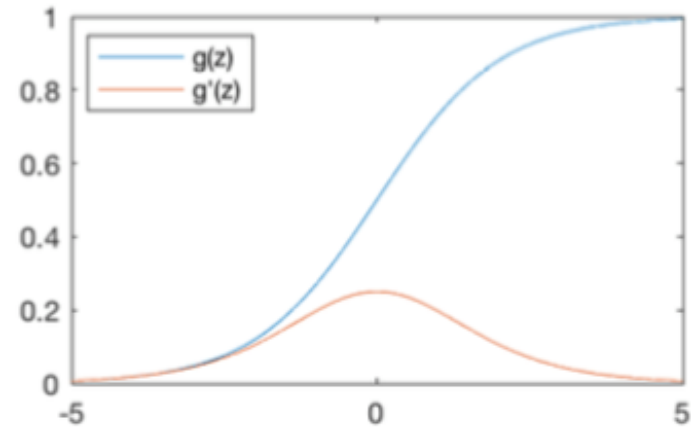


$$z_i^{(k)} = g\left(\sum_j W_{i,j}^{(k-1,k)} z_j^{(k-1)}\right)$$

**g = nonlinear activation function**

# Common Activation Functions

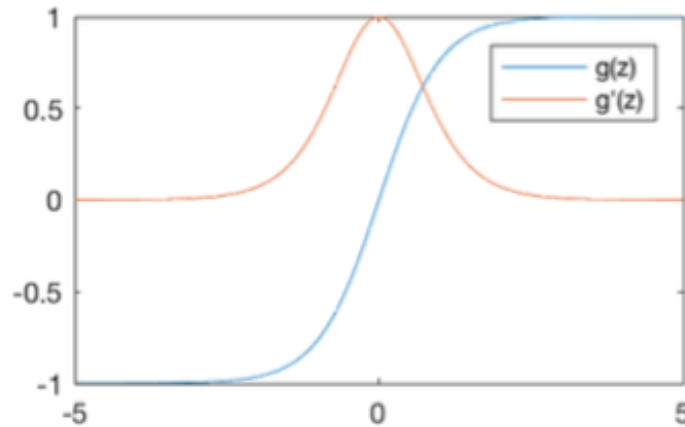
Sigmoid Function



$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g'(z) = g(z)(1 - g(z))$$

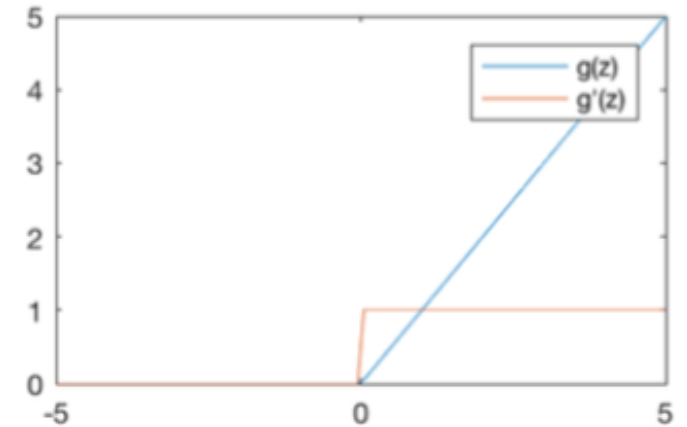
Hyperbolic Tangent



$$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$g'(z) = 1 - g(z)^2$$

Rectified Linear Unit (ReLU)



$$g(z) = \max(0, z)$$

$$g'(z) = \begin{cases} 1, & z > 0 \\ 0, & \text{otherwise} \end{cases}$$



# Deep Neural Network: Also Learn the Features!

- Training the deep neural network is just like logistic regression:

$$\max_w ll(w) = \max_w \sum_i \log P(y^{(i)} | x^{(i)}; w)$$

just  $w$  tends to be a much, much larger vector 😊

→ just run gradient ascent

+ stop when log likelihood of hold-out data starts to decrease

# Neural Networks Properties

---

- Theorem (Universal Function Approximators). A two-layer neural network with a sufficient number of neurons can approximate any continuous function to any desired accuracy.
- Practical considerations
  - Can be seen as learning the features
  - Large number of neurons
    - Danger for overfitting
    - (hence early stopping!)

# Fun Neural Net Demo Site

---

- Demo-site:
  - <http://playground.tensorflow.org/>

# How about computing all the derivatives?

- Derivatives tables:

$$\frac{d}{dx}(a) = 0$$

$$\frac{d}{dx}(x) = 1$$

$$\frac{d}{dx}(au) = a \frac{du}{dx}$$

$$\frac{d}{dx}(u + v - w) = \frac{du}{dx} + \frac{dv}{dx} - \frac{dw}{dx}$$

$$\frac{d}{dx}(uv) = u \frac{dv}{dx} + v \frac{du}{dx}$$

$$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{1}{v} \frac{du}{dx} - \frac{u}{v^2} \frac{dv}{dx}$$

$$\frac{d}{dx}(u^n) = nu^{n-1} \frac{du}{dx}$$

$$\frac{d}{dx}(\sqrt{u}) = \frac{1}{2\sqrt{u}} \frac{du}{dx}$$

$$\frac{d}{dx}\left(\frac{1}{u}\right) = -\frac{1}{u^2} \frac{du}{dx}$$

$$\frac{d}{dx}\left(\frac{1}{u^n}\right) = -\frac{n}{u^{n+1}} \frac{du}{dx}$$

$$\frac{d}{dx}[f(u)] = \frac{d}{du}[f(u)] \frac{du}{dx}$$

$$\frac{d}{dx}[\ln u] = \frac{d}{dx}[\log_e u] = \frac{1}{u} \frac{du}{dx}$$

$$\frac{d}{dx}[\log_a u] = \log_a e \frac{1}{u} \frac{du}{dx}$$

$$\frac{d}{dx}e^u = e^u \frac{du}{dx}$$

$$\frac{d}{dx}a^u = a^u \ln a \frac{du}{dx}$$

$$\frac{d}{dx}(u^v) = vu^{v-1} \frac{du}{dx} + \ln u \cdot u^v \frac{dv}{dx}$$

$$\frac{d}{dx} \sin u = \cos u \frac{du}{dx}$$

$$\frac{d}{dx} \cos u = -\sin u \frac{du}{dx}$$

$$\frac{d}{dx} \tan u = \sec^2 u \frac{du}{dx}$$

$$\frac{d}{dx} \cot u = -\csc^2 u \frac{du}{dx}$$

$$\frac{d}{dx} \sec u = \sec u \tan u \frac{du}{dx}$$

$$\frac{d}{dx} \csc u = -\csc u \cot u \frac{du}{dx}$$

# How about computing all the derivatives?

---

- But neural net  $f$  is never one of those?
  - No problem: CHAIN RULE:

If  $f(x) = g(h(x))$

Then  $f'(x) = g'(h(x))h'(x)$

→ Derivatives can be computed by following well-defined procedures

# Automatic Differentiation

---

- Automatic differentiation software
  - e.g. Theano, TensorFlow, PyTorch, Chainer
  - Only need to program the function  $g(x,y,w)$
  - Can automatically compute all derivatives w.r.t. all entries in  $w$
- Need to know this exists
- How this is done? -- outside of scope of CSE573

# Summary of Key Ideas

- Optimize probability of label given input  $\max_w ll(w) = \max_w \sum_i \log P(y^{(i)} | x^{(i)}; w)$
- Continuous optimization
  - Gradient ascent:
    - Compute steepest uphill direction = gradient (= just vector of partial derivatives)
    - Take step in the gradient direction
    - Repeat (until held-out data accuracy starts to drop = “early stopping”)
- Deep neural nets
  - Last layer = still logistic regression
  - Now also many more layers before this last layer
    - = computing the features
    - → the features are learned rather than hand-designed
  - Universal function approximation theorem
    - If neural net is large enough
    - Then neural net can represent any continuous mapping from input to output with arbitrary accuracy
    - But remember: need to avoid overfitting / memorizing the training data → early stopping!
  - Automatic differentiation gives the derivatives efficiently (how? = outside of scope of 573)

# How well does it work?

---



# How well does it work?

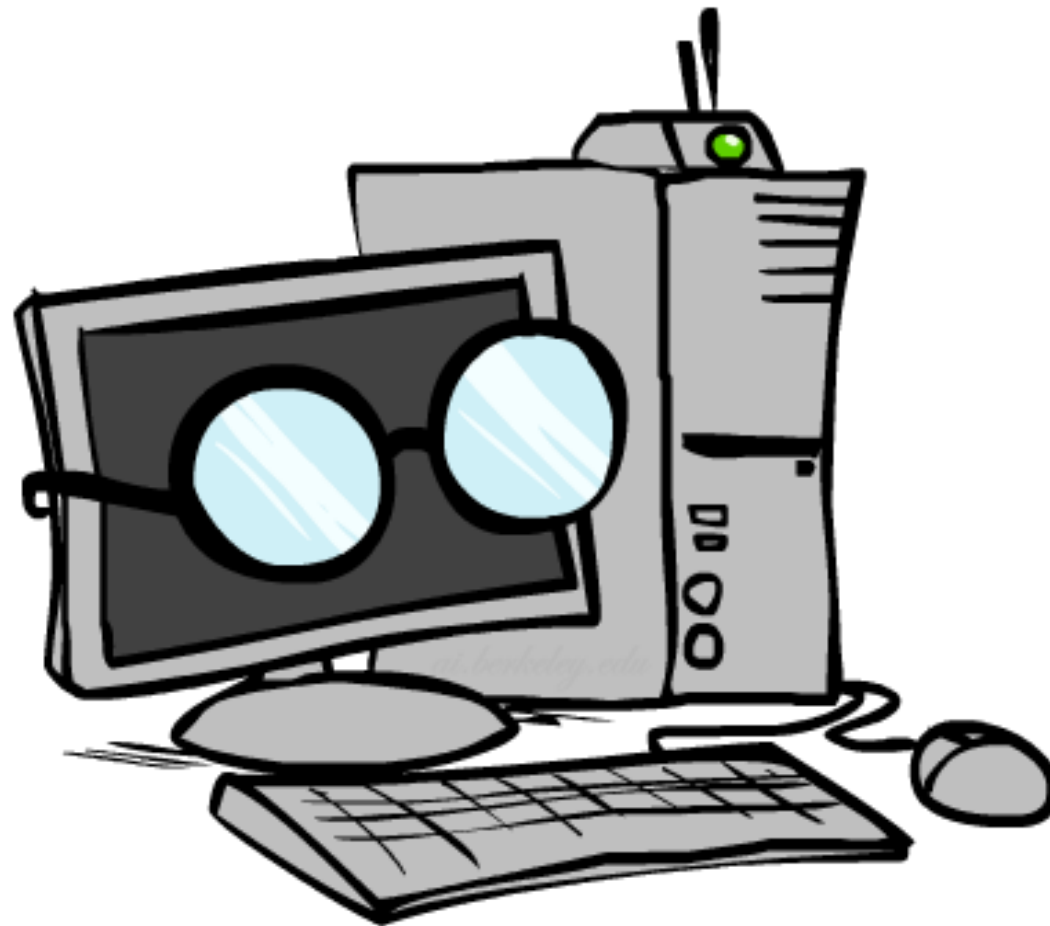
---

# Next: More Neural Net Applications!

---

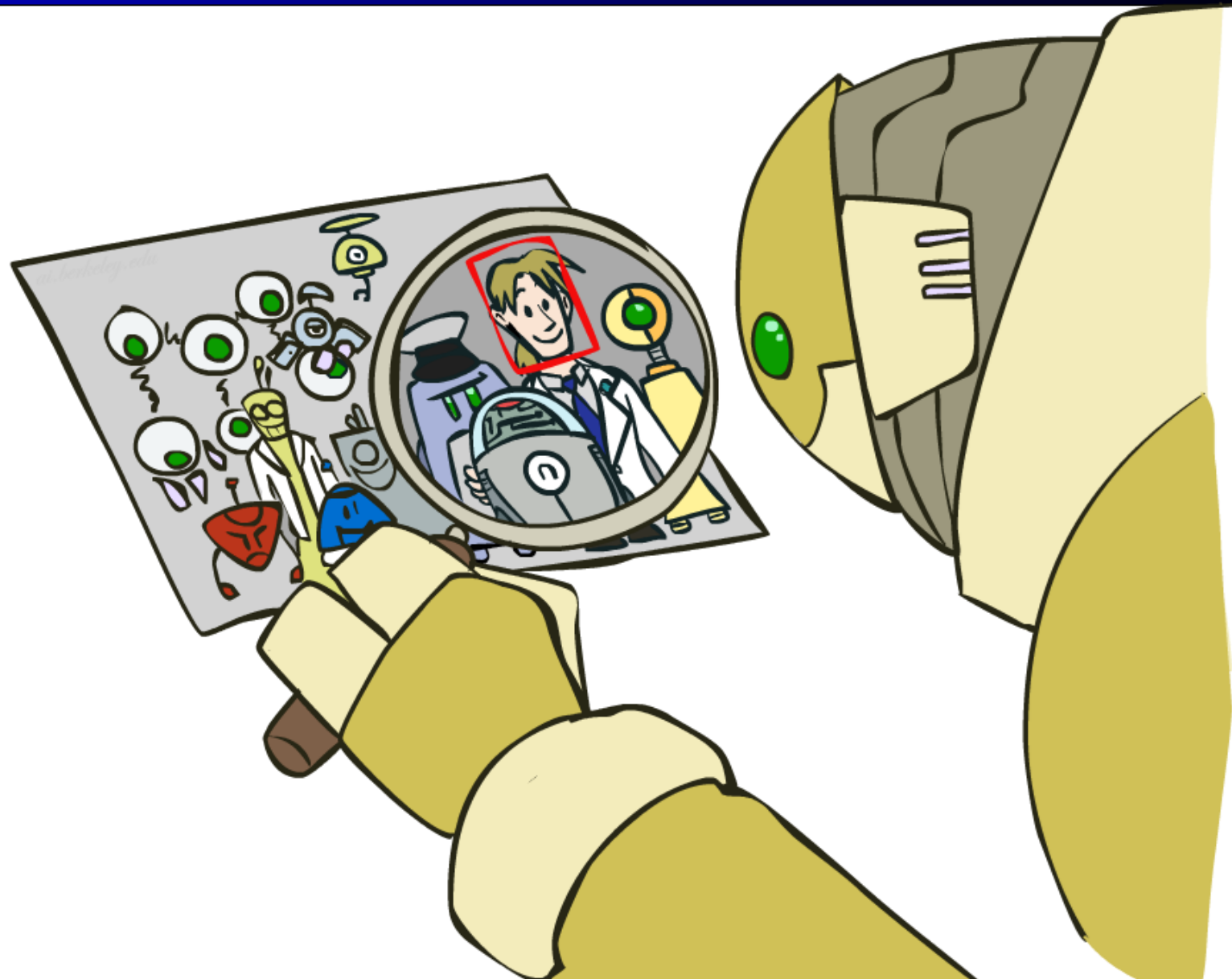
# Computer Vision

---

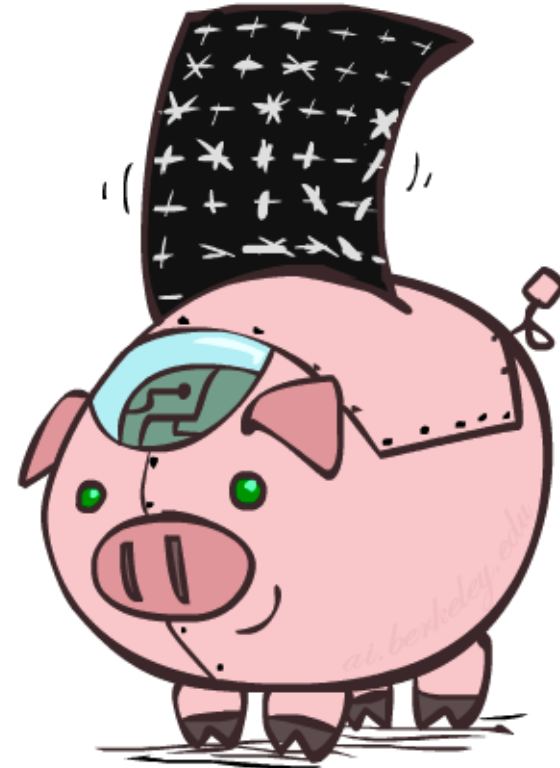
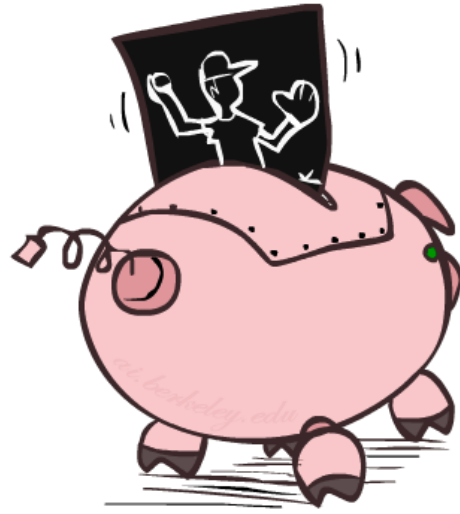


# Object Detection

---

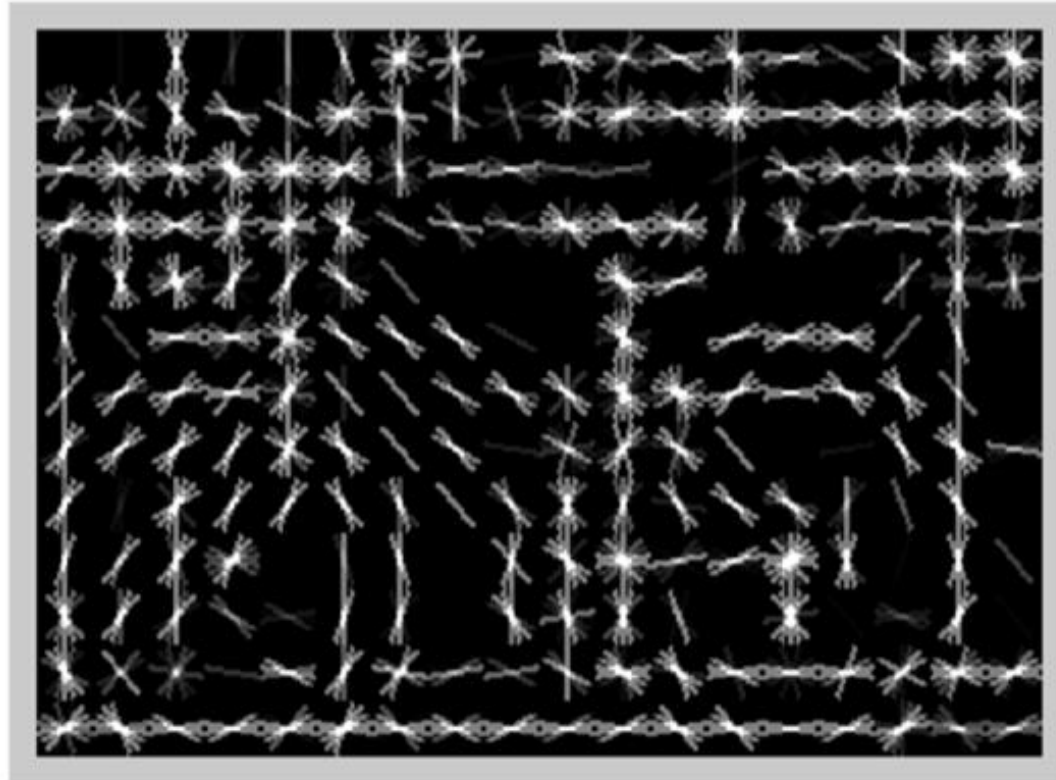


# Manual Feature Design



# Features and Generalization

---

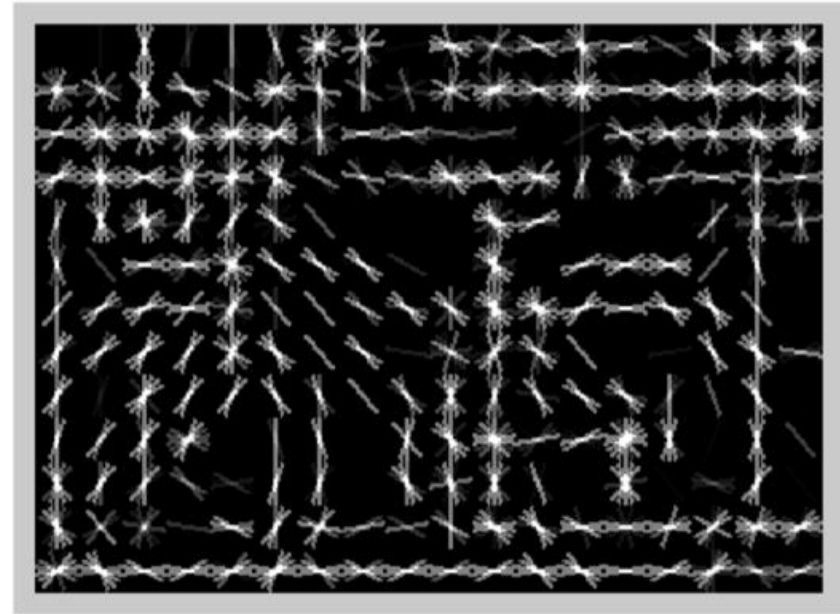


# Features and Generalization

---



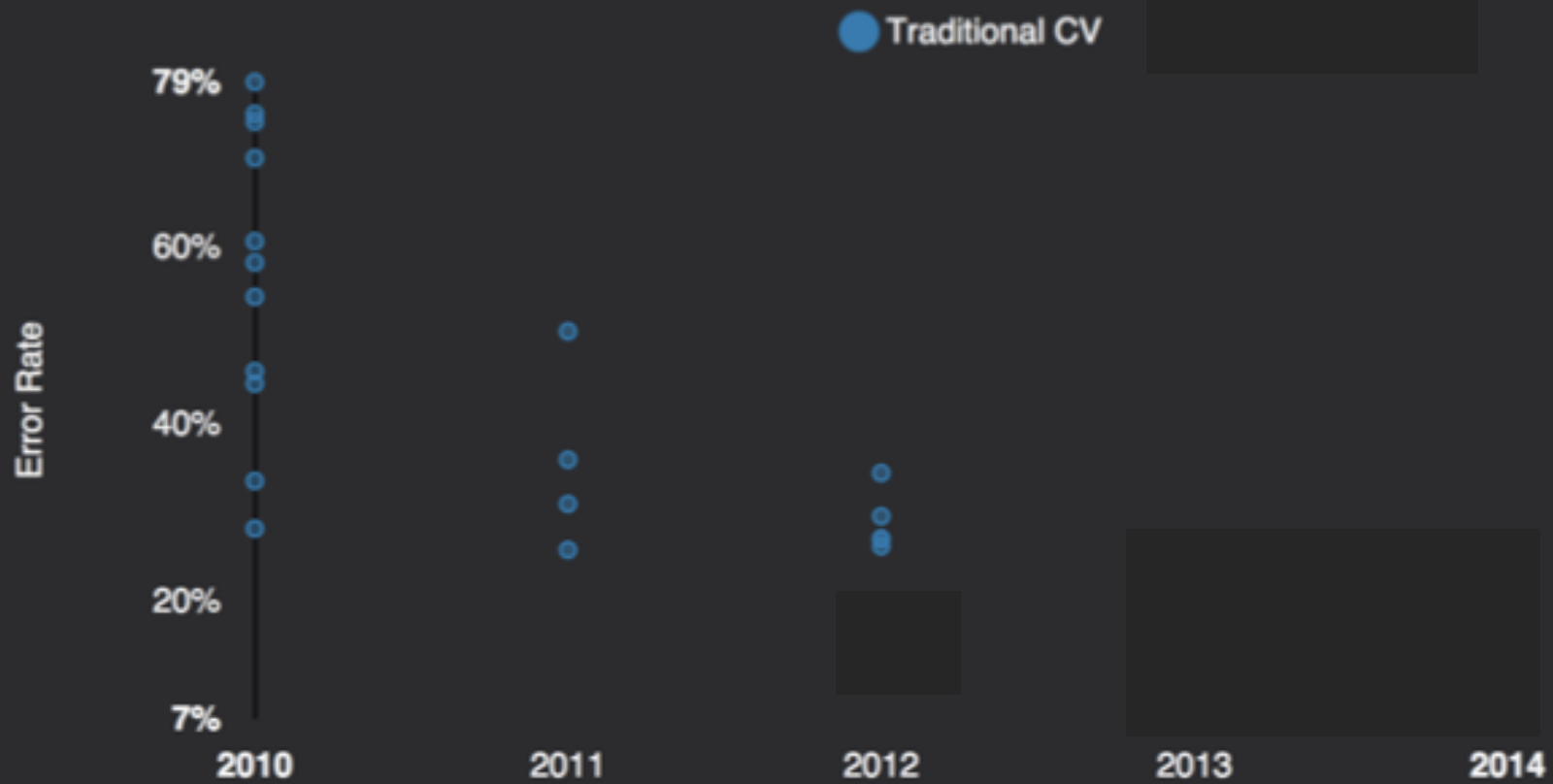
Image



HoG

# Performance

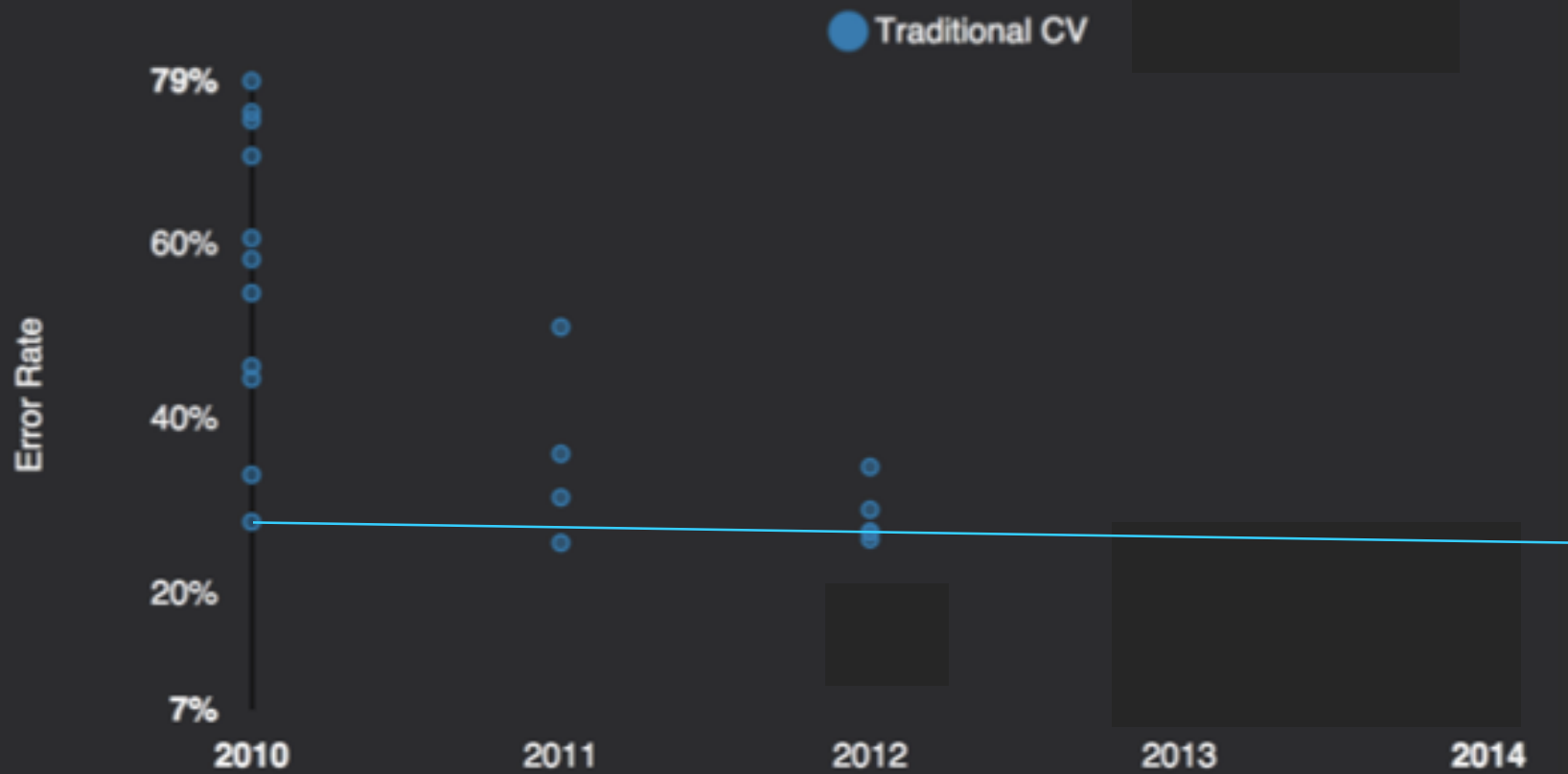
## ImageNet Error Rate 2010-2014





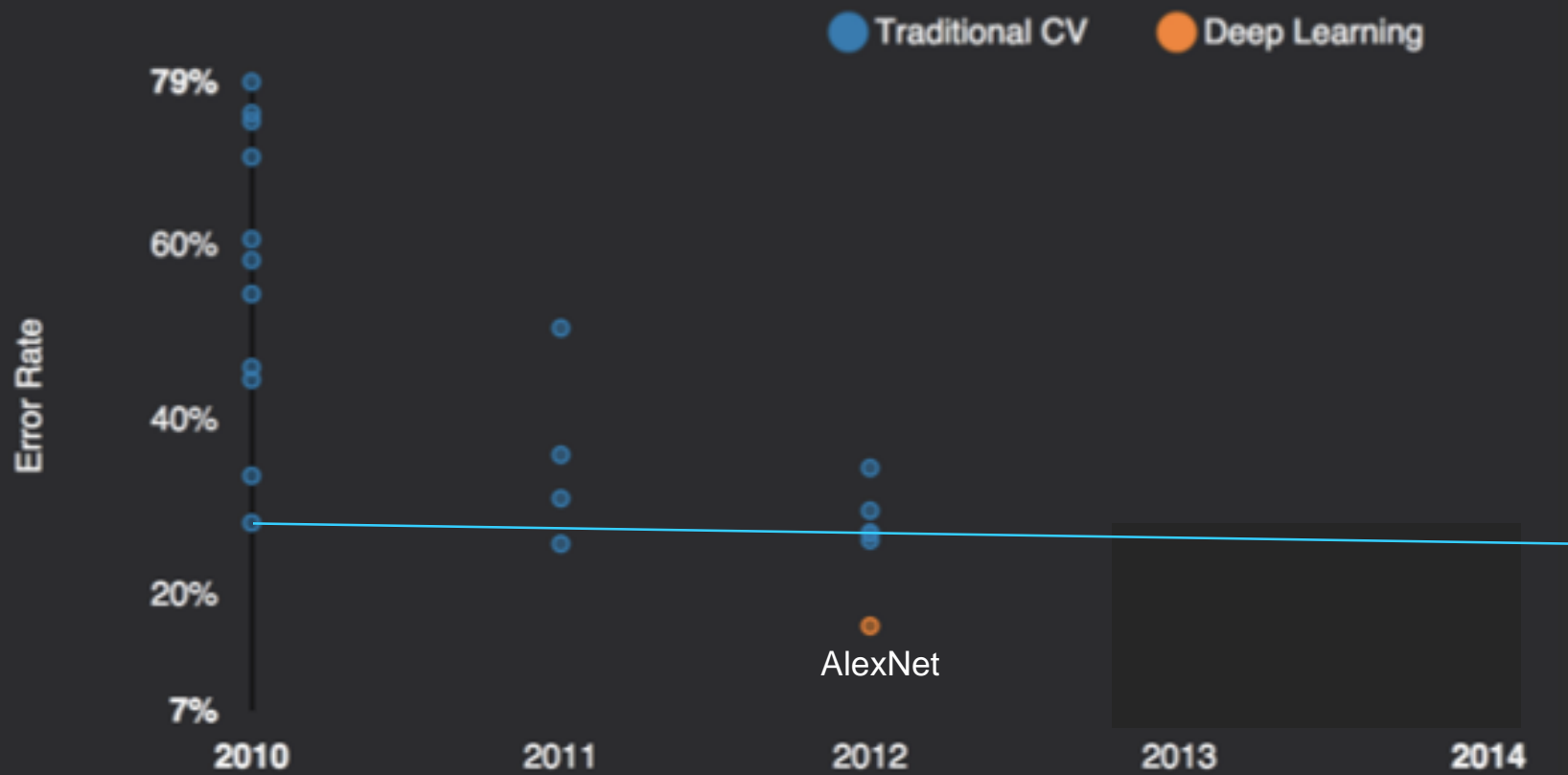
# Performance

## ImageNet Error Rate 2010-2014



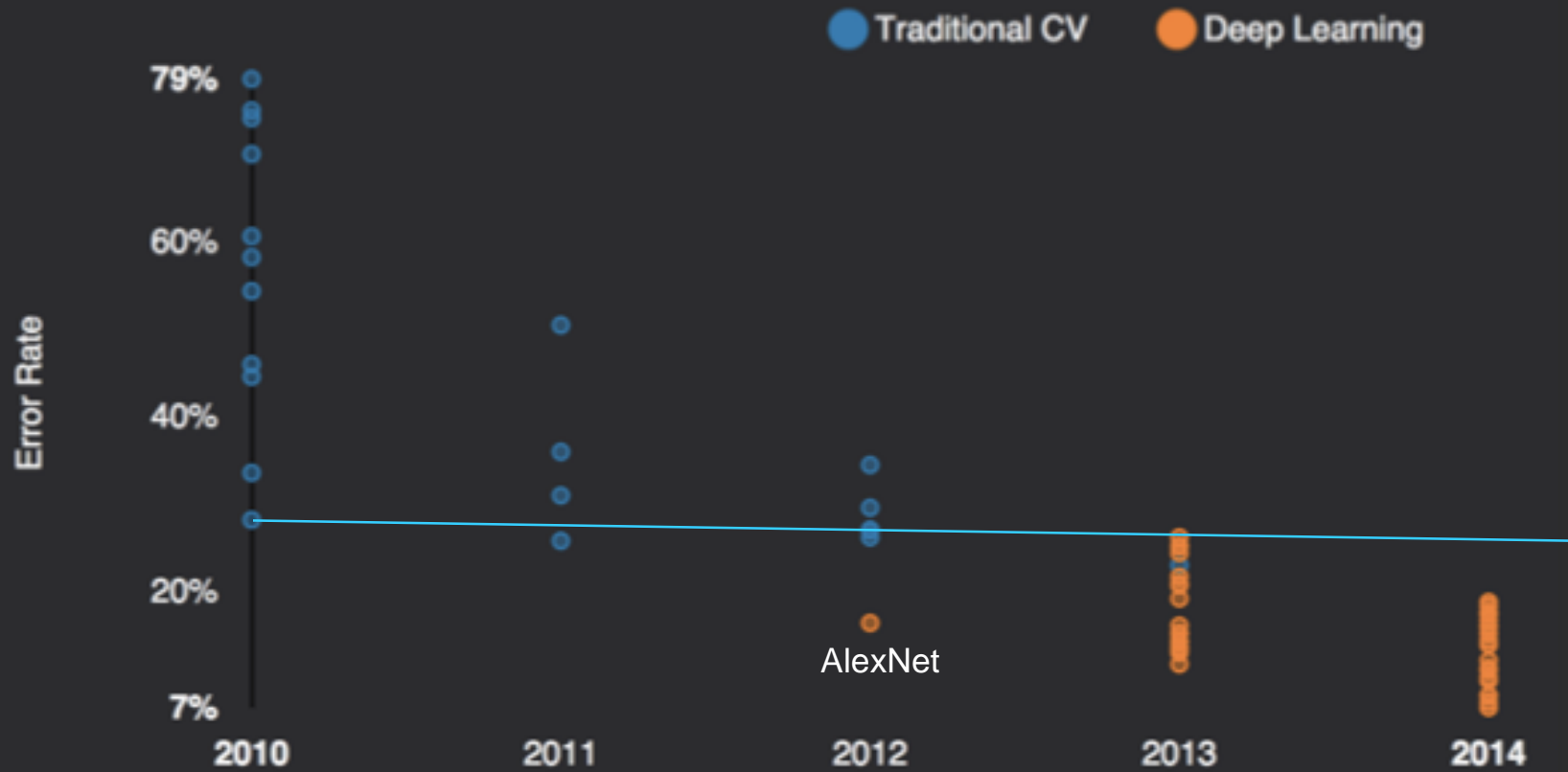
# Performance

## ImageNet Error Rate 2010-2014



# Performance

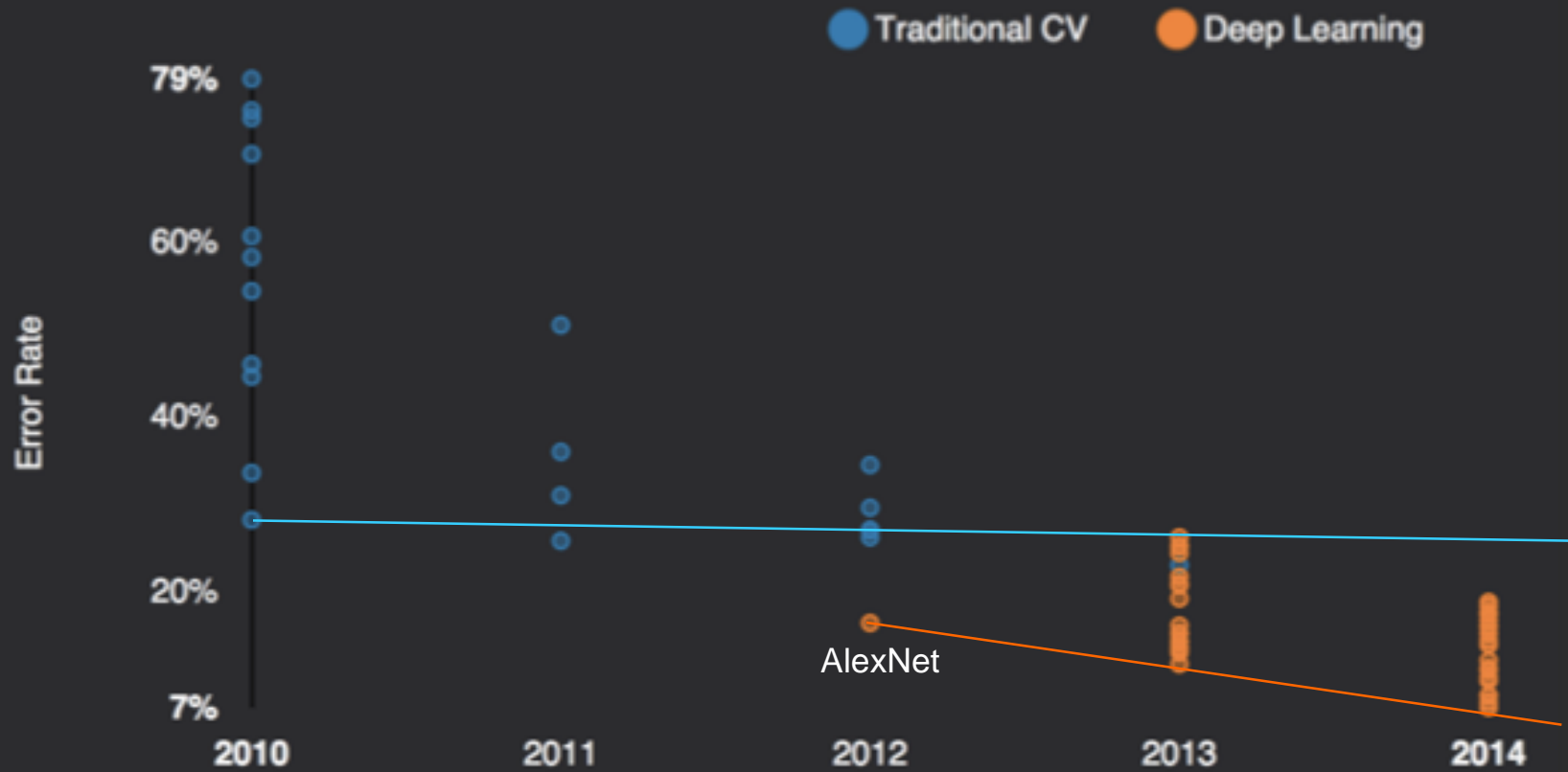
## ImageNet Error Rate 2010-2014



graph credit Matt Zeiler, Clarifai

# Performance

## ImageNet Error Rate 2010-2014



# Papers With Code: ImageNet

Leaderboard

Dataset

View

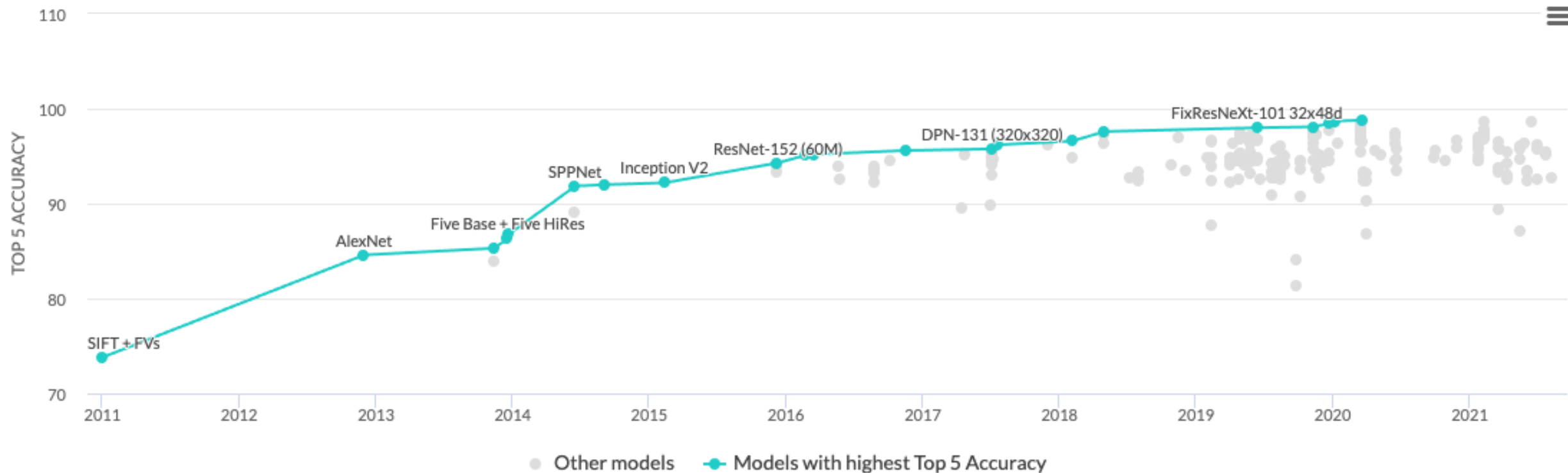
Top 5 Accuracy

by

Date

for

All models



# MS COCO Image Captioning Challenge



"man in black shirt is playing guitar."



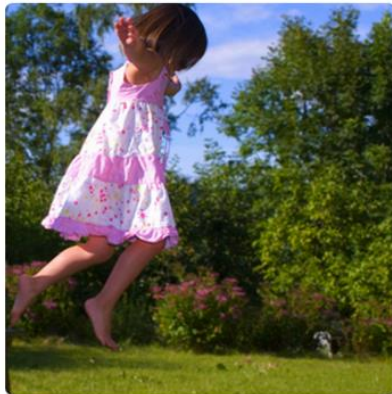
"construction worker in orange safety vest is working on road."



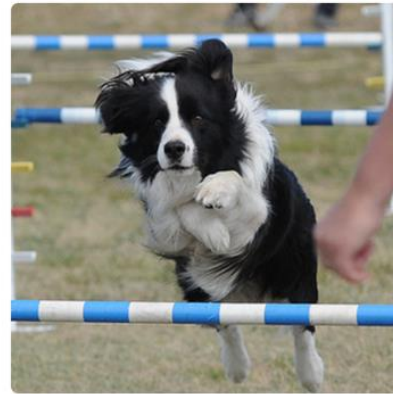
"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



"young girl in pink shirt is swinging on swing."



"man in blue wetsuit is surfing on wave."



# Visual QA Challenge

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, Devi Parikh



What vegetable is on the plate?

Neural Net: broccoli

Ground Truth: broccoli



What color are the shoes on the person's feet ?

Neural Net: brown

Ground Truth: brown



How many school busses are there?

Neural Net: 2

Ground Truth: 2



What sport is this?

Neural Net: baseball

Ground Truth: baseball



What is on top of the refrigerator?

Neural Net: magnets

Ground Truth: cereal



What uniform is she wearing?

Neural Net: shorts

Ground Truth: girl scout



What is the table number?

Neural Net: 4

Ground Truth: 40



What are people sitting under in the back?

Neural Net: bench

Ground Truth: tent



# Image Segmentation





# Question Answering

Context

Super Bowl 48 was an American football game to determine the champion of the National Football League (NFL) for the 2013 season. The National Football Conference champions Seattle Seahawks defeated the American Football Conference champions Denver Broncos. The Seahawks defeated the Broncos 43—8, the largest margin victory for an underdog and tied the third largest point differential overall (35) in Super Bowl history with Super Bowl XXVII (1993). It was the first time the winning scored over 40 points, while holding their opponent to under 10.

Question

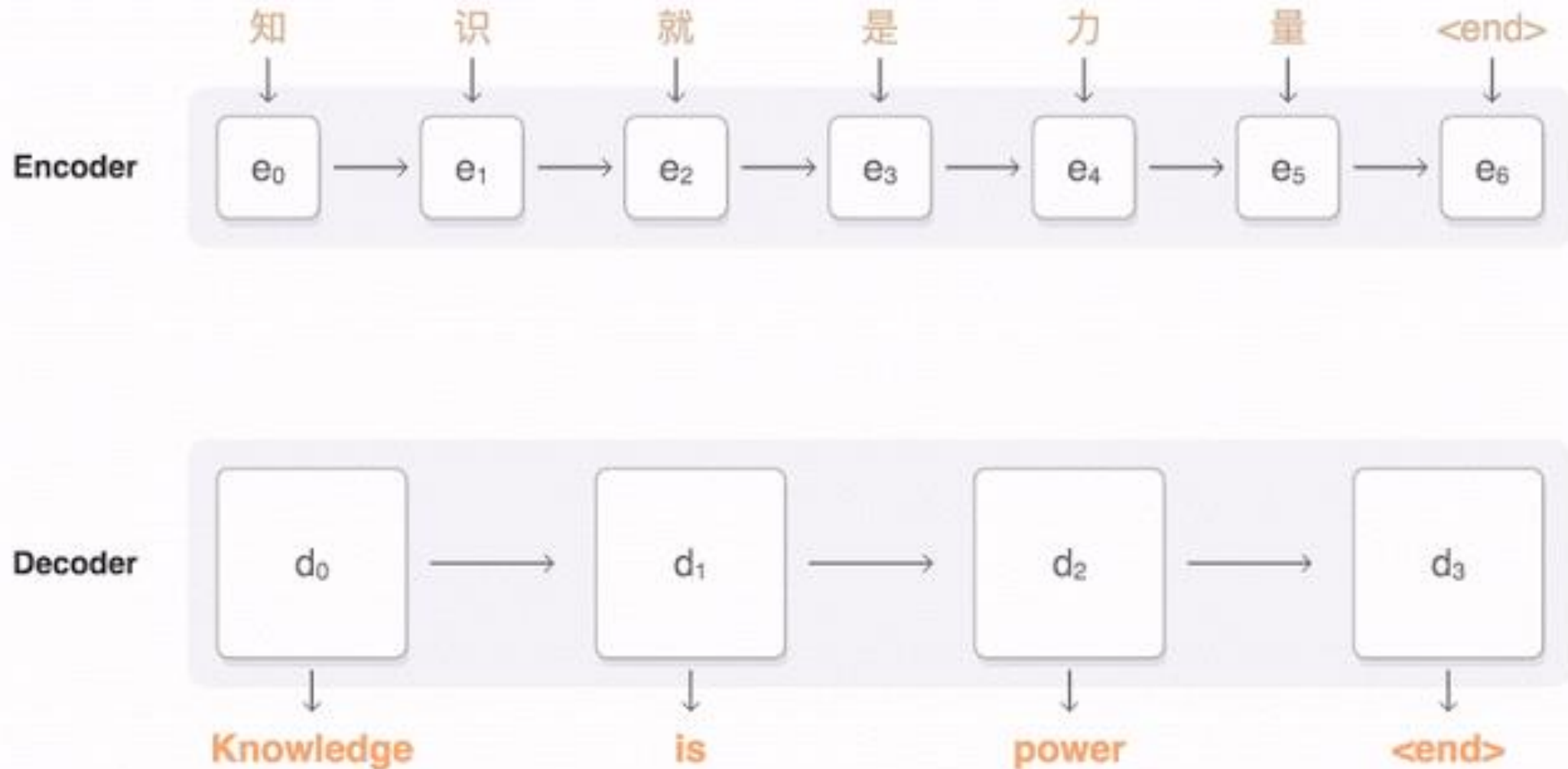
Which NFL team represented the NFC at Super Bowl 48?

Answer

Seattle Seahawks

# Machine Translation

Google Neural Machine Translation (in production)



# Pipeline Approach for Question Answering

- Feature engineering
- Classifying phrases

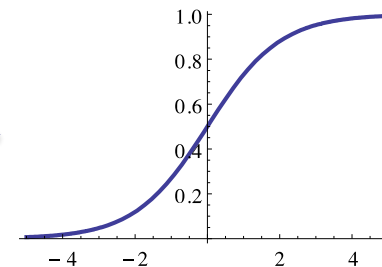
Super Bowl 48 was an American football game to determine the champion of the National Football League (NFL) for the 2013 season. The National Football Conference champions Seattle Seahawks defeated the American Football Conference champions Denver Broncos. The Seahawks defeated the Broncos 43—8, the largest margin victory for an underdog and tied the third largest point differential overall (35) in Super Bowl history with Super Bowl XXVII (1993). It was the first time the winning scored over 40 points, while holding their opponent to under 10.

Which NFL team represented the NFC at Super Bowl 48?

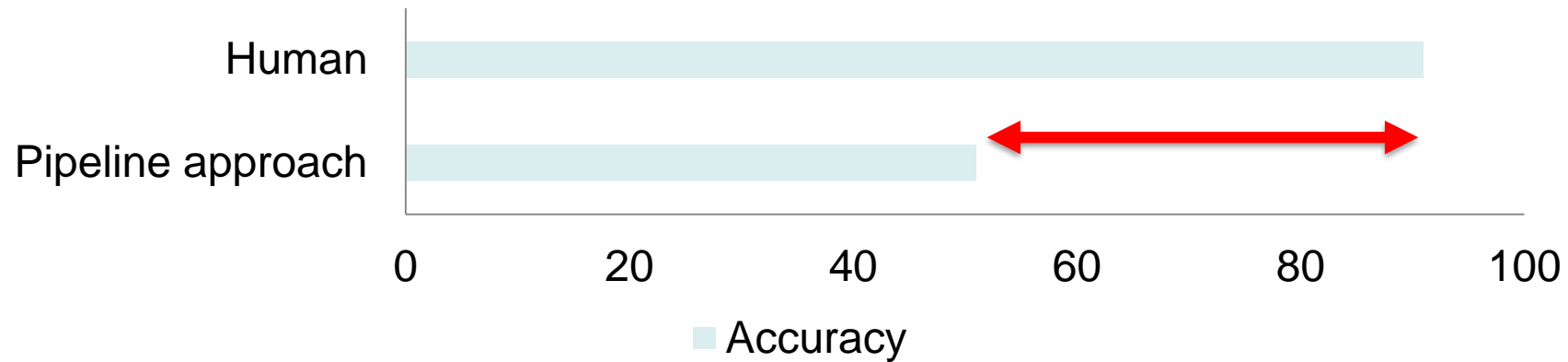
words, types, frequencies  
dependency relations



$$f_1, f_2, \dots, f_n$$



# Pipeline Approach Results

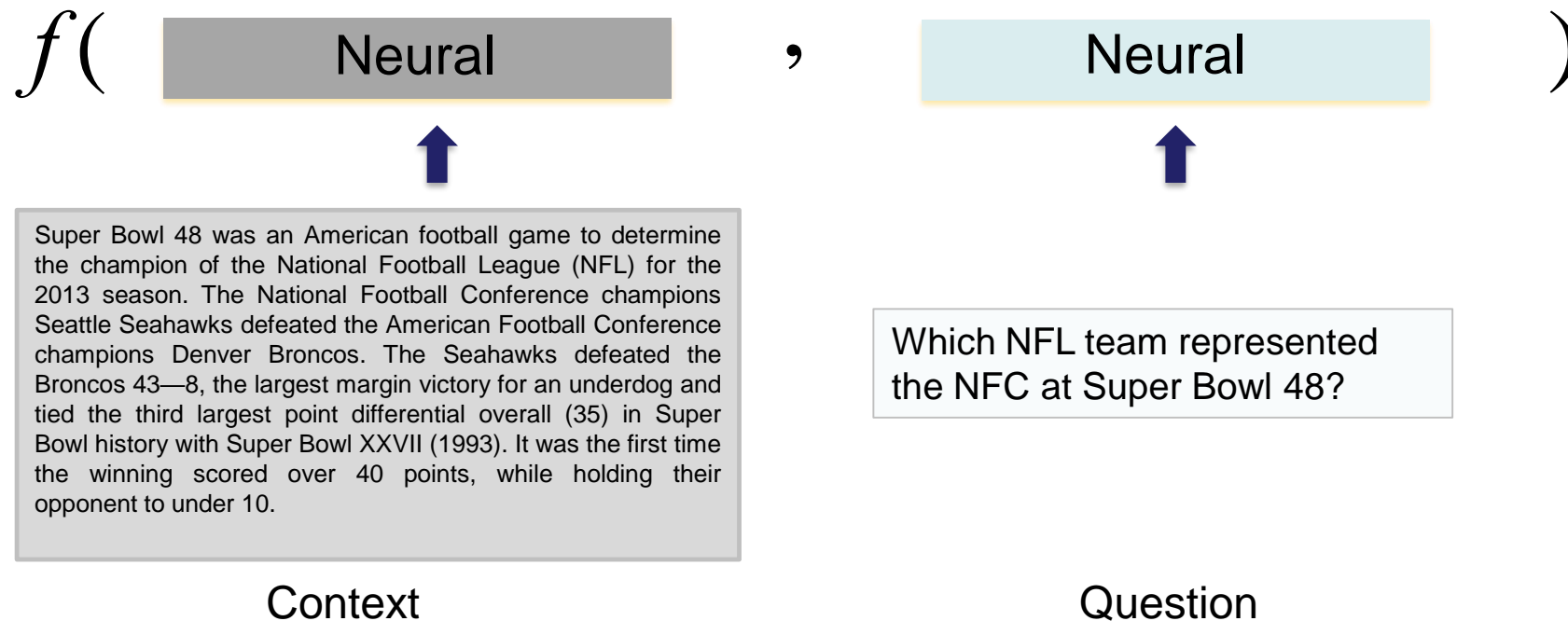


- Dataset: Stanford Question Answering Dataset (SQuAD) [Rajpurkar et al 2016]:
  - 100k Wikipedia documents with question
- Accuracy: percentage of correctly predicted phrases

# Neural Approach

[ICLR'17]

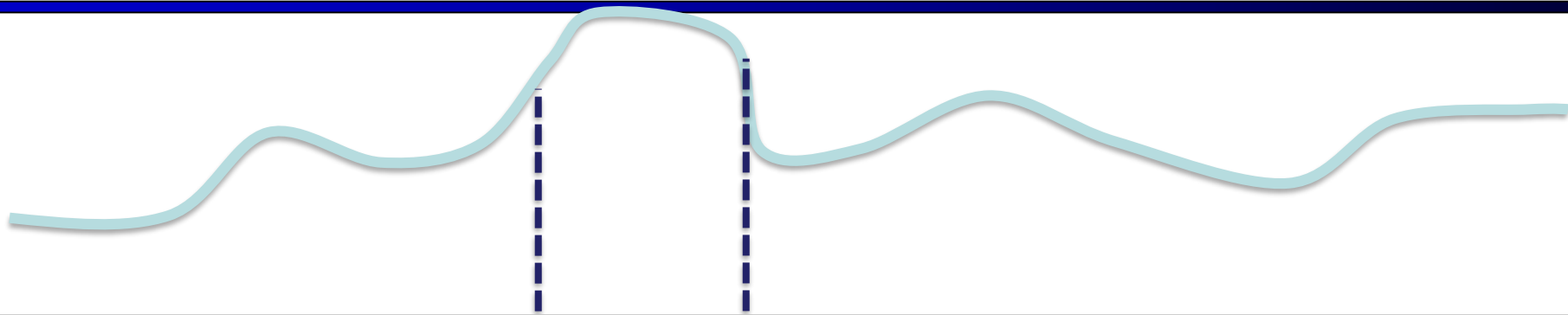
Find a function that assigns a high score to the the correct answer given the context and question



Super Bowl 48 was an American football game to determine the champion of the National Football League (NFL) for the 2013 season. The National Football Conference champions Seattle Seahawks defeated the American Football Conference champions Denver Broncos. The Seahawks defeated the Broncos 43—8, the largest margin victory for an underdog and tied the third largest point differential overall (35) in Super Bowl history with Super Bowl XXVII (1993). It was the first time the winning scored over 40 points, while holding their opponent to under 10.

Which NFL team represented the NFC at Super Bowl 48?

Seattle Seahawks



The National Football Conference champions Seattle Seahawks defeated the American Football Conference champions Denver Broncos.

$f($  **Encoding** , **Encoding** )

Super Bowl 48 was an American football game to determine the champion of the National Football League (NFL) for the 2013 season. The National Football Conference champions Seattle Seahawks defeated the American Football Conference champions Denver Broncos. The Seahawks defeated the Broncos 43—8, the largest margin victory for an underdog and tied the third largest point differential overall (35) in Super Bowl history with Super Bowl XXVII (1993). It was the first time the winning scored over 40 points, while holding their opponent to under 10.

Which NFL team represented the NFC at Super Bowl 48?

Context

Question

# Question Answering Leaderboard

Jan 1, 2017

## Test Set Leaderboard

Since the release of our dataset (and paper), the community has made rapid progress! Here are the ExactMatch (EM) and F1 scores of the best models evaluated on the test and development sets of v1.1.

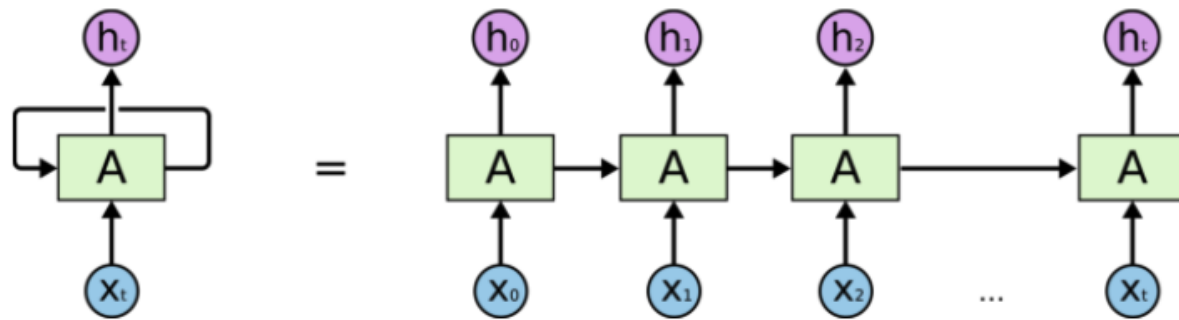
Rank	Model	Test EM	Test F1
1	BiDAF (ensemble) Allen Institute for AI & University of Washington (Seo et al. '16)	73.3	81.1
2	Dynamic Coattention Networks (ensemble) Salesforce Research (Xiong & Zhong et al. '16)	71.6	80.4
2	r-net (ensemble) Microsoft Research Asia	72.1	79.7
4	r-net (single model) Microsoft Research Asia	68.4	77.5
5	BiDAF (single model) Allen Institute for AI & University of Washington (Seo et al. '16)	68.0	77.3
5	Multi-Perspective Matching (ensemble) IBM Research	68.2	77.2

March 8, 2021

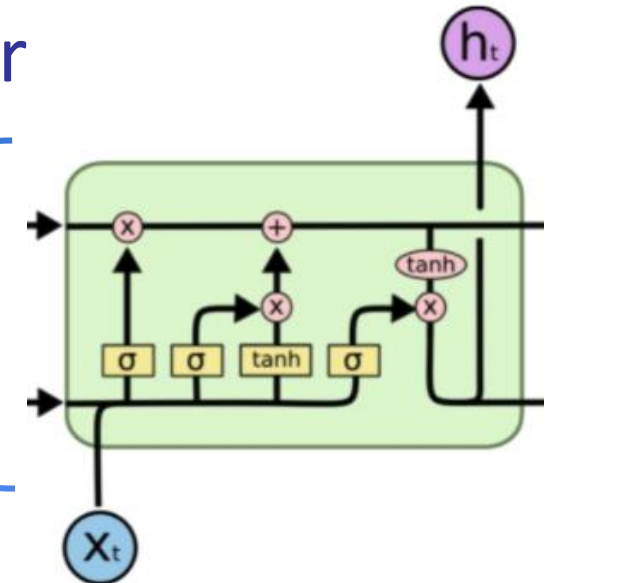
Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Feb 21, 2021	FPNet (ensemble) Ant Service Intelligence Team	90.871	93.183
2 Feb 24, 2021	IE-Net (ensemble) RICOH_SRCB_DML	90.758	93.044
3 Apr 06, 2020	SA-Net on Albert (ensemble) QIANXIN	90.724	93.011
4 May 05, 2020	SA-Net-V2 (ensemble) QIANXIN	90.679	92.948
4 Apr 05, 2020	Retro-Reader (ensemble) Shanghai Jiao Tong University <a href="http://arxiv.org/abs/2001.09694">http://arxiv.org/abs/2001.09694</a>	90.578	92.978
4 Feb 05, 2021	FPNet (ensemble) YuYang	90.600	92.899
5 Dec 01, 2020	EntitySpanFocusV2 (ensemble) RICOH_SRCB_DML	90.521	92.824
5 Jul 31, 2020	ATRLP+PV (ensemble) Hithink RoyalFlush	90.442	92.877
5 May 04, 2020	ELECTRA+ALBERT+EntitySpanFocus (ensemble) SRCB_DML	90.442	92.839

# Neural Networks for Natural Language Processing

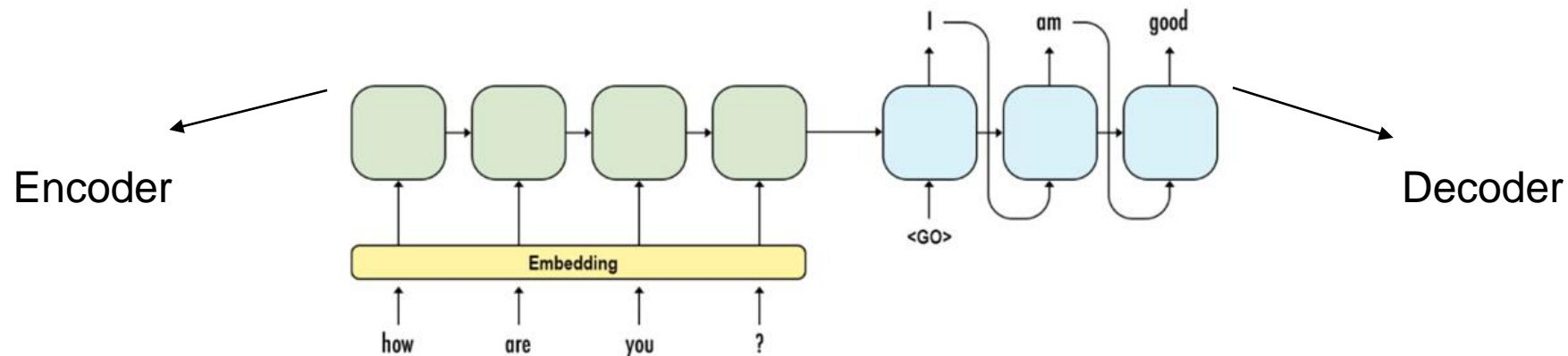
**Recurrent Neurons** to encode and decode (i.e. generate sequential data like **TEXT**).



An unrolled recurrent neural network.



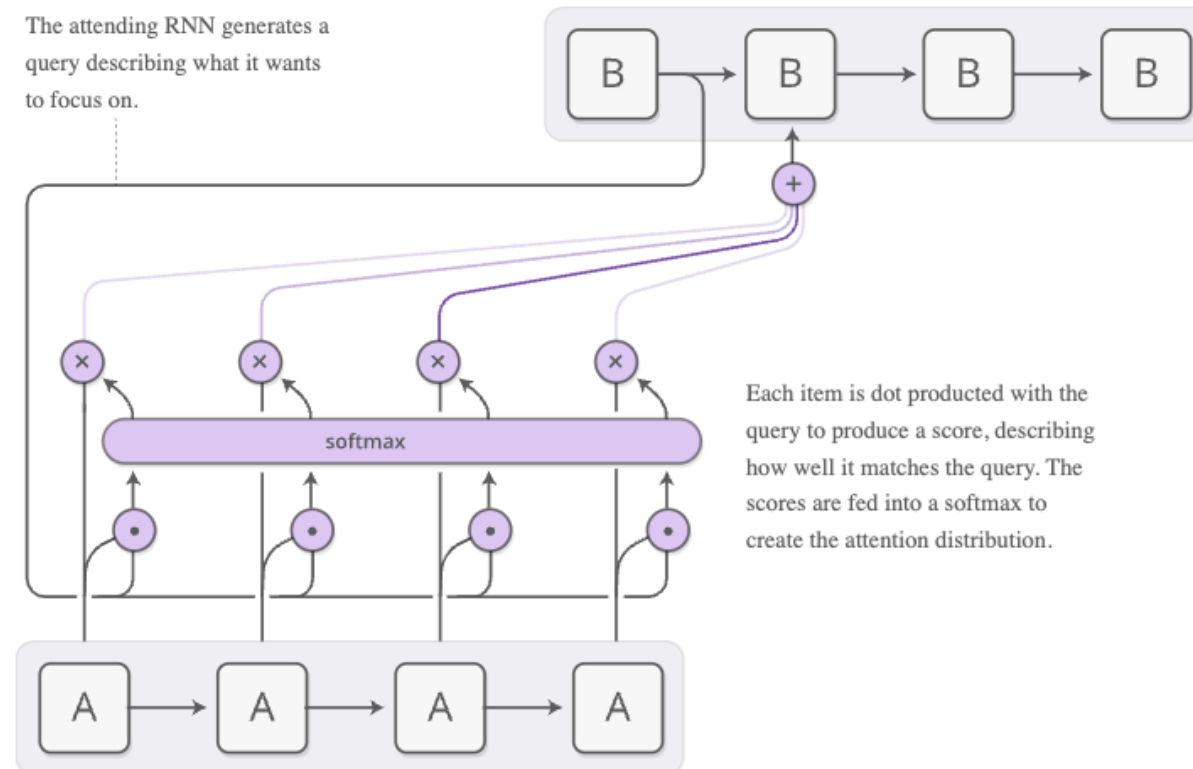
Long Short-Term Memory (LSTM) cell





# Attending to Input

Attention helps resolve the **Vanishing Gradient Problem** that recurrent neural networks suffer over LONG SEQUENCES. At any time step, the model can decide which tokens to pay attention to from other time-steps.

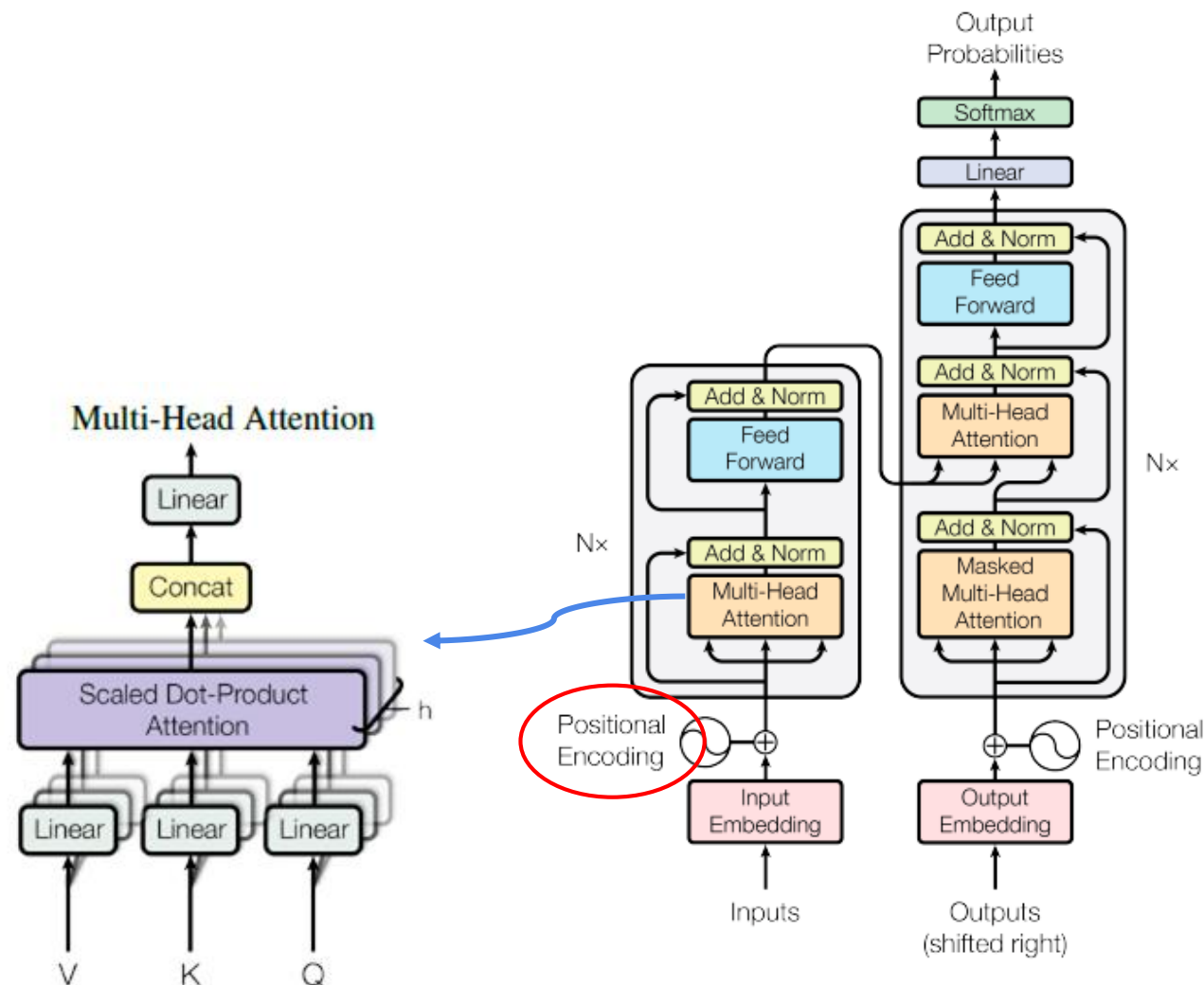


# Attention is all you need!

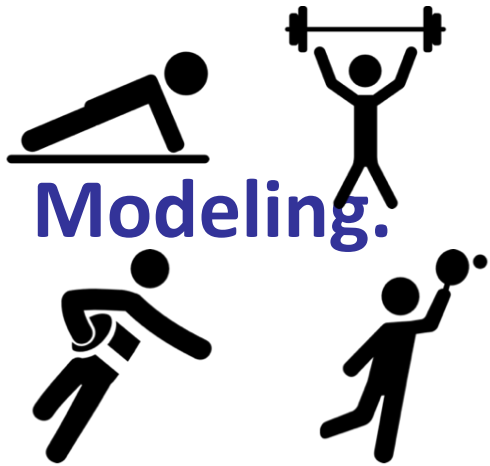
Turns out you only need attention, and can get rid of the recurrent neurons entirely!

**Transformers** : Interleaving attention layers and fully-connected layers, which can be computed **parallelly** over the sequence, instead of recurrently.

**Positional Embeddings** : Encode Sequence Information



# Pretrain-then-finetune paradigm



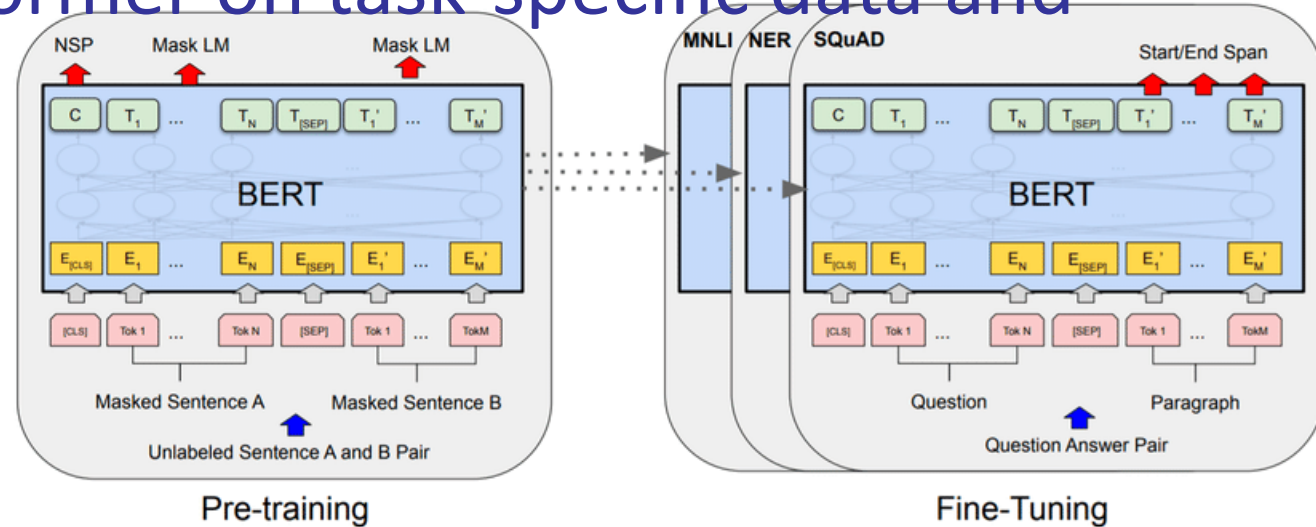
Modeling.

Pre-train transformer on **Masked Language**

Finetune transformer on task-specific data and

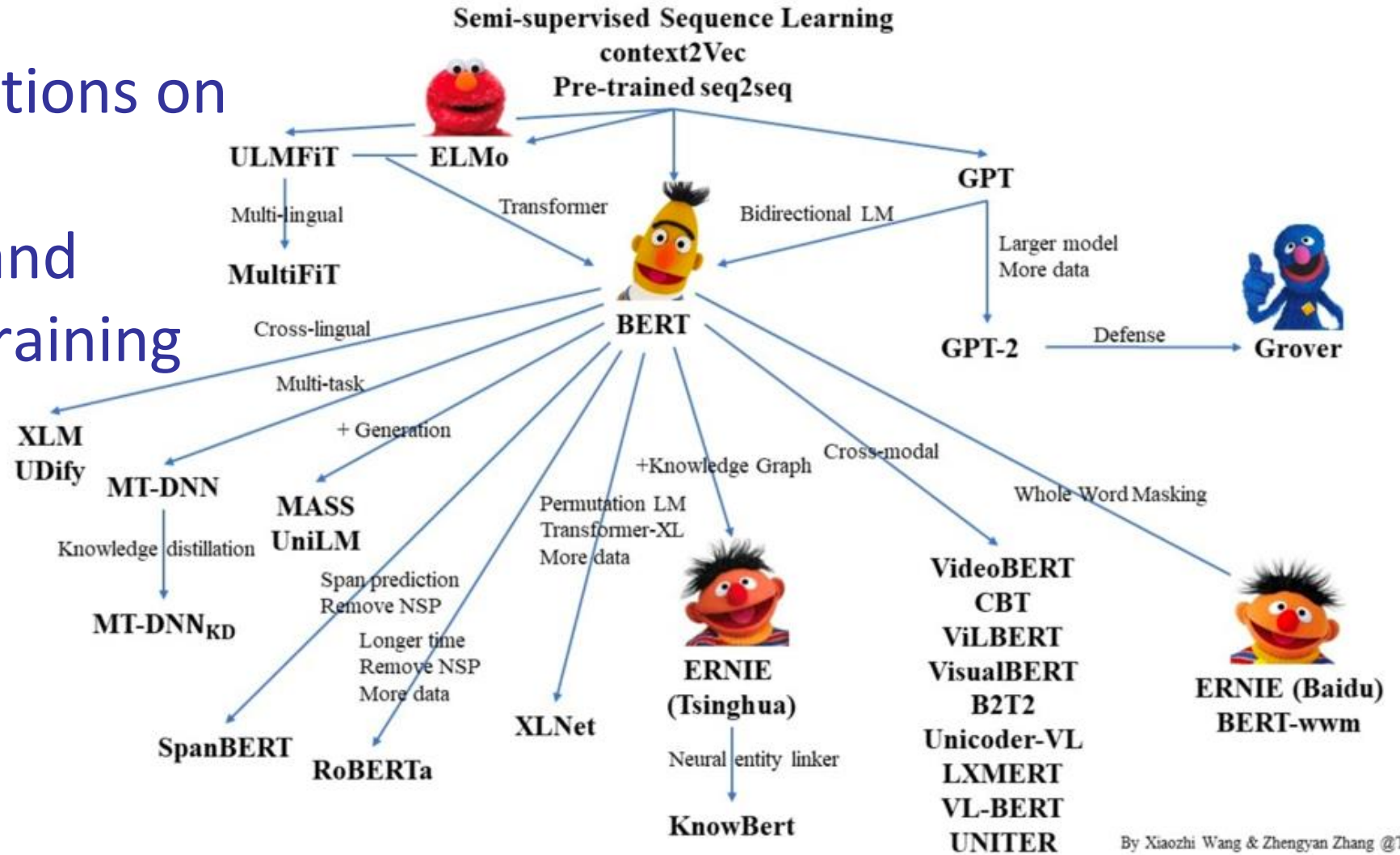
labels.

Randomly masked A quick [MASK] fox jumps over the [MASK] dog  
↓  
Predict A quick brown fox jumps over the lazy dog



# BERT and Family

Different Variations on Transformer architectures and different pre-training tasks



# NLP Tasks and Benchmarks

---

## 🔗 Language **Understanding** tasks

- ✂ GLUE/SuperGLUE Benchmarks

- ✂ Natural Language Entailment, Paraphrase detection, Sentiment/review classification

- ✂ Question Answering, Reading Comprehension

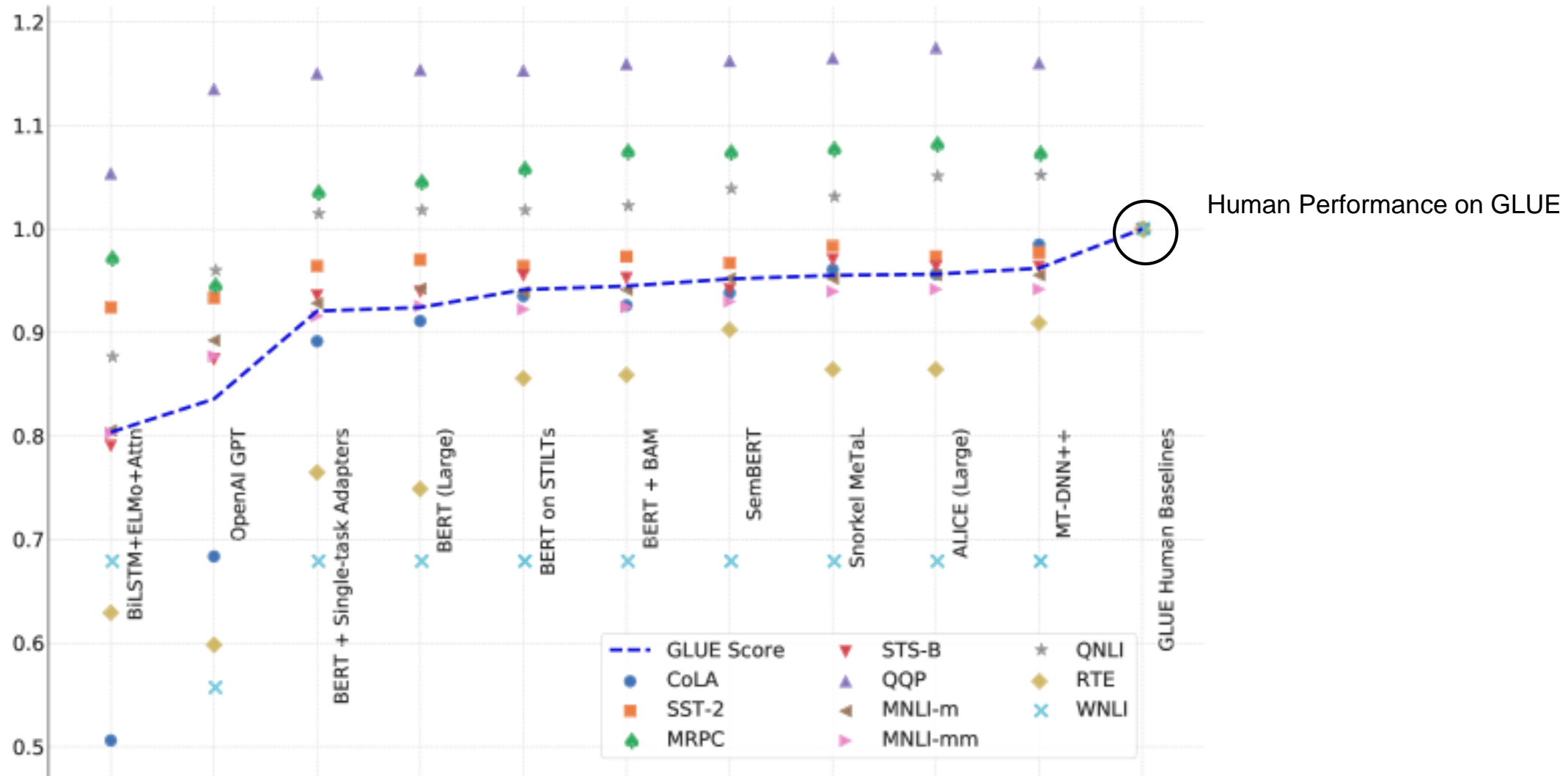
## 🔗 Language **Generating** tasks

- ✂ Machine Translation

- ✂ Long-text summarization

🔗 Dialogue Systems: Interactive systems that have to **understand** humans and **generate** responses

# Pretrained Models (BERT) on GLUE Benchmarks



# Massive Pre-trained models are few-shot learners!

## (GPT-3)

175B GPT-3 can work without fine-tuning, when it is shown sample **demonstrations** for a task:

### Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1	Translate English to French:	← task description
2	sea otter => loutre de mer	← examples
3	peppermint => menthe poivrée	← examples
4	plush girafe => girafe peluche	← examples
5	cheese => .....	← prompt

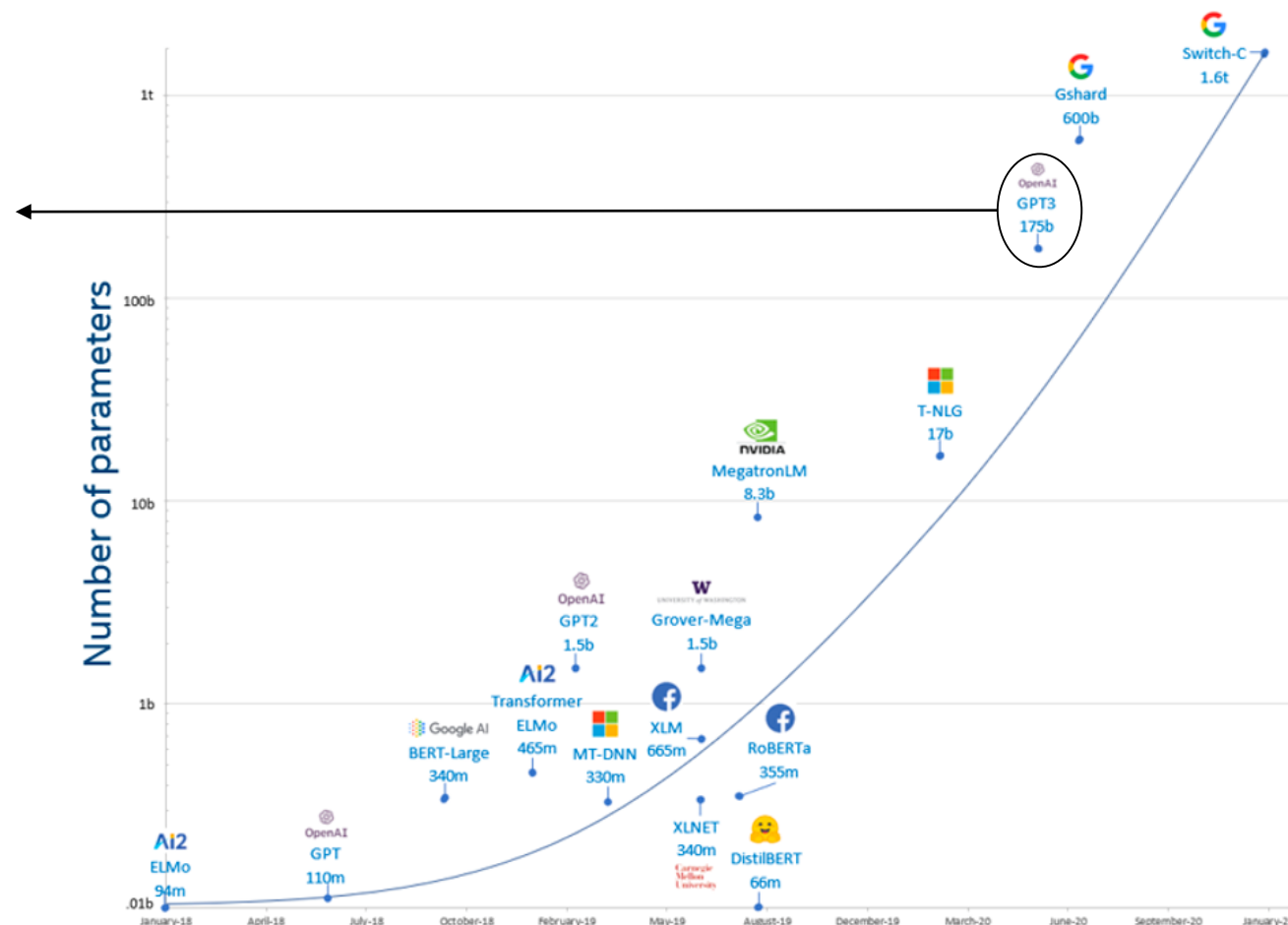


Figure 1: Exponential growth of number of parameters in DL models



# CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning

Pranav Rajpurkar\*, Jeremy Irvin\*, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew P. Lungren, Andrew Y. Ng

We develop an algorithm that can detect pneumonia from chest X-rays at a level exceeding practicing radiologists.

Chest X-rays are currently the best available method for diagnosing pneumonia, playing a crucial role in clinical care and epidemiological studies. Pneumonia is responsible for more than 1 million hospitalizations and 50,000 deaths per year in the US alone.

[READ OUR PAPER](#)





# Google and DeepMind are using AI to predict the energy output of wind farms

*To help make that energy more valuable to the power grid*

By [Nick Statt](#) | [@nickstatt](#) | Feb 26, 2019, 2:42pm EST

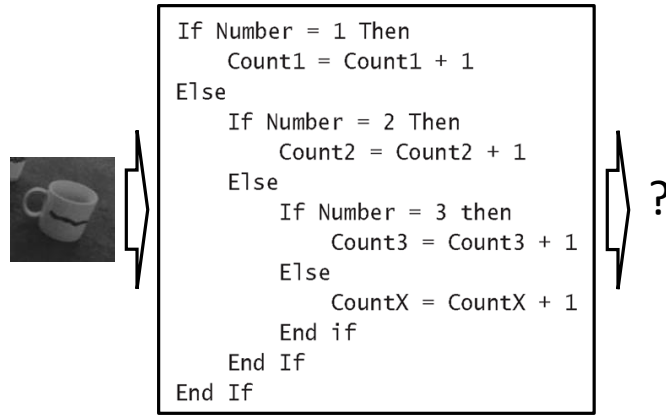
[f](#) [t](#) [SHARE](#)



Google [announced today](#) that it has made energy produced by wind farms more viable using the artificial intelligence software of its London-based subsidiary DeepMind. By using DeepMind's machine learning algorithms to predict the wind output from the farms Google uses for its green energy initiatives, the company says it can now schedule set deliveries of energy output, which are more valuable to the grid than standard, non-time-based deliveries.

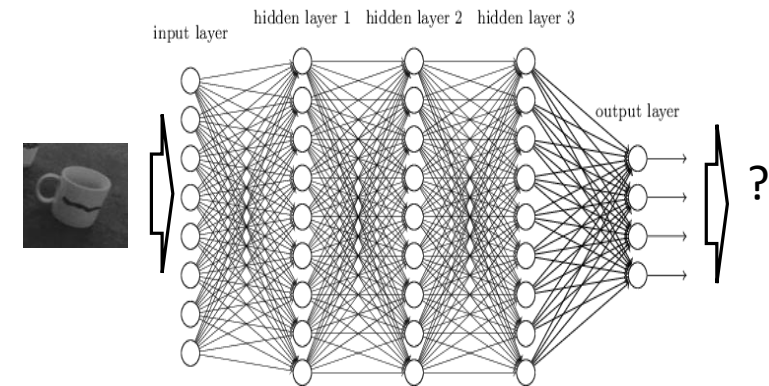
# Change in Programming Paradigm!

Traditional Programming:  
program by writing lines of code



Poor performance on AI problems

Deep Learning ("Software 2.0"):  
program by providing data



Success!