Neural Networks 2

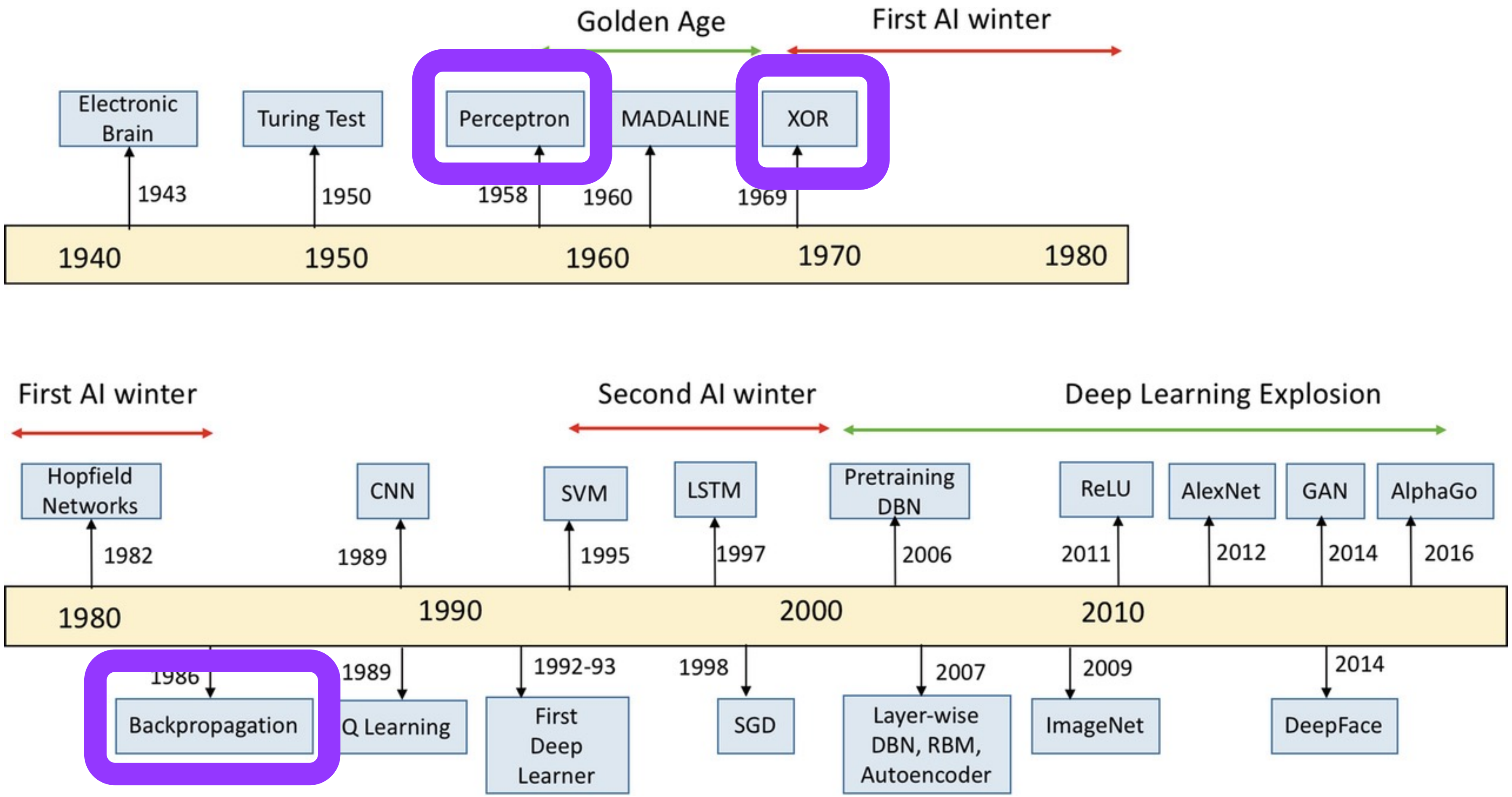
466/566 Fall 2022

Some of these slides based on content from Seyong Kim and Nando de Freitas
Nando’s youtube lectures (which are are really great!) See last slide for link

Administrivia

- Monday is a holiday (no office hours, no lab)
- Thursday of next week (Oct 13): **no class**
- Midterm is Oct 20 (two weeks from today)
- We will have some time for **review in class on the 18th**
  - Come with questions, I will not be preparing anything

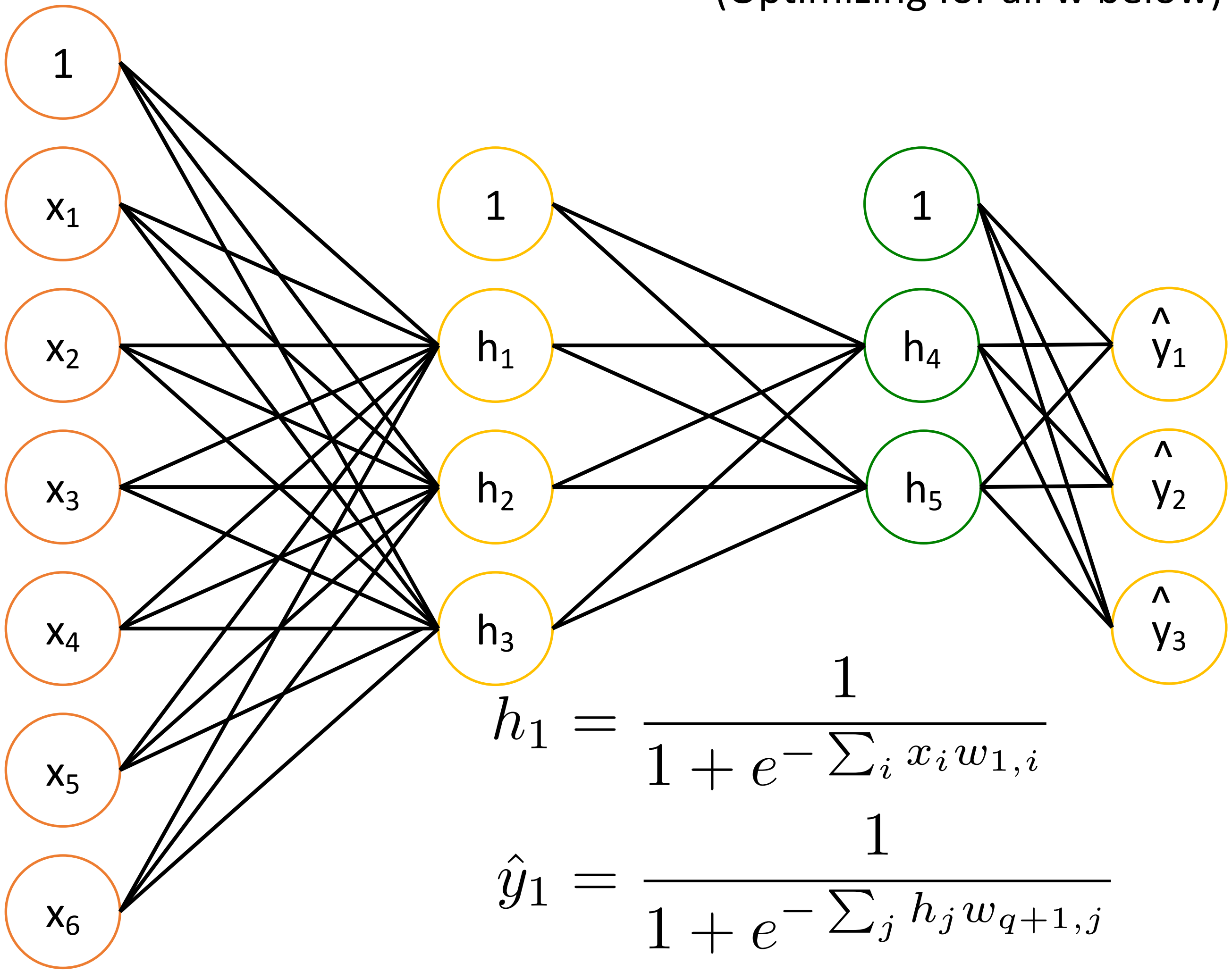
A little of the history of NNs



Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

Now: Back Prop

(Optimizing for all w below)



$$h_1 = \frac{1}{1 + e^{-\sum_i x_i w_{1,i}}}$$

$$\hat{y}_1 = \frac{1}{1 + e^{-\sum_j h_j w_{q+1,j}}}$$

q is total # of hidden nodes across all layers

Backprop Algorithm

1. Randomly initialize weights (w)
2. Repeat until convergence
    1. For each (batch, mini-batch) in data
        1. For each data point in batch
            1. Forward pass (calculate all intermediate values h, s)
            2. Backwards pass compute gradient of loss wrt all w
        2. Average gradient over data points in batch
        3. Update w

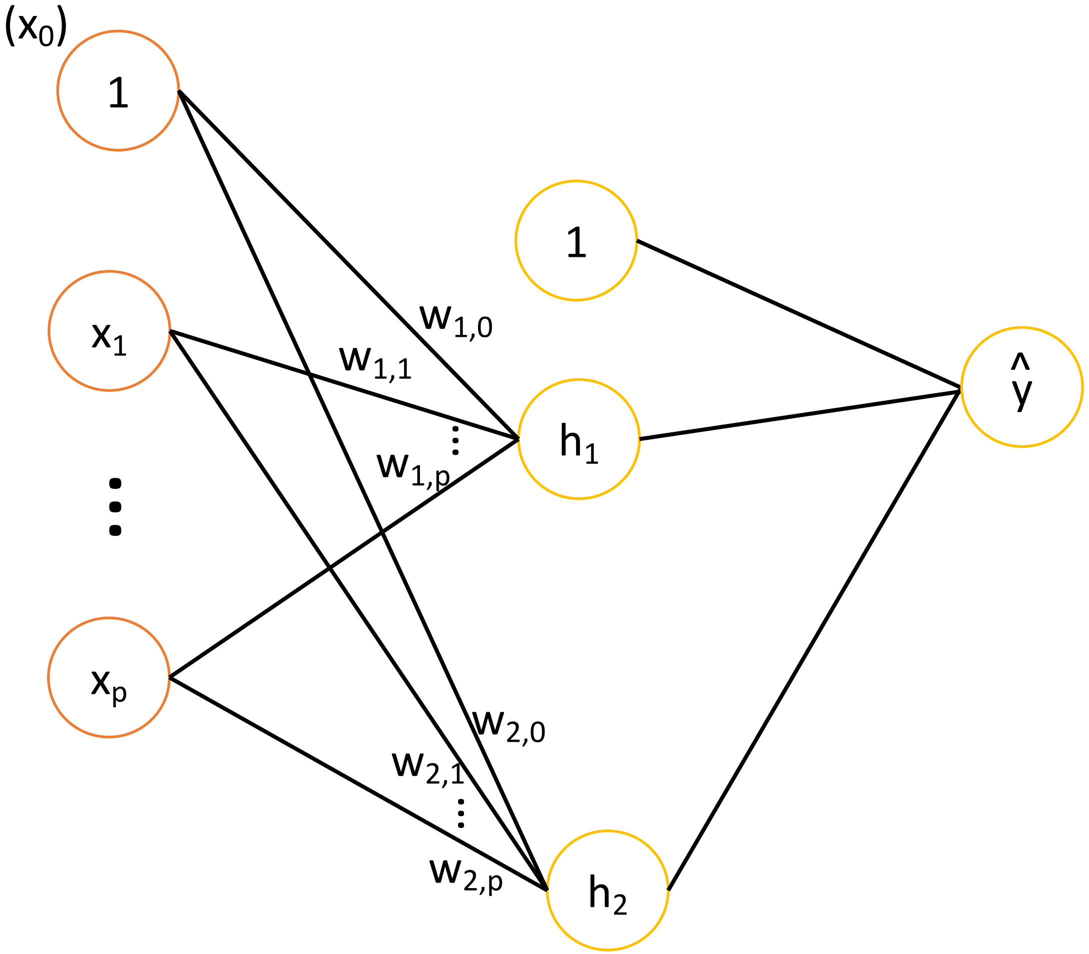
Backprop takes advantage of the fact that many of the value you need for each gradient are computed in the forward pass, or as part of another gradient.

Our example for in class

$$h_1 = \frac{1}{1+e^{-\sum_i x_i w_{1,i}}}$$

$$\hat{y} = \sum_j h_j w_{3,j}$$

(Optimizing for all w below)



$$\hat{y} = \sum_j h_j w_{3,j}$$

$$h_1 = \sigma(s_1)$$

$$s_1 = \sum_{i=0}^{p} w_{1,i} x_i$$

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

Techniques for speeding up SGD

Momentum

- Sometimes SGD updates produce oscillations, overstepping the most optimal path.
- Momentum mixes a fraction of the last update into the current update, which helps control oscillations

A lovely tutorial with math for these concepts: https://ruder.io/optimizing-gradient-descent/

Momentum

- $v_t = \gamma v_{t-1} + \eta \nabla_w J(w)$
- $w = w - v_t$

- The update is a mix of the regular SGD update ($\eta \nabla_w J(w)$) and the last update (γv_{t-1}) γ usually around 0.9

A lovely tutorial with math for these concepts: https://ruder.io/optimizing-gradient-descent/

Nesterov accelerated gradient (NAG)

- Momentup update can be rewritten
  - $w=\underbrace{w-\gamma v_{t-1}}_{\text{Lookahead}}-\eta\nabla_w J(w))$
- NAG notes that we have some of the info we need to compute part of that update ahead of time
  - $v_t=\gamma v_{t-1}+\eta\nabla_w J(w-\gamma v_{t-1})$
  - $w=w-v_t$

Adagrad

- Adapt the learning rate based on the frequency of a feature
  - Features that appear often have weights that are updated with a smaller step size
  - The *learning rate* changes based on magnitude of the past updates

  - $g_t = \nabla_w J(w_{t,i})$
  - $w_{t+1} = w_t - \dfrac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$

    $\boxed{\text{Without the denominator, this is just the regular SGD update}}$
  - G_t is a diagonal matrix where each diagonal element i,i is the sum of the squares of the gradients w.r.t. element i of w up to time step t

Adagrad

- $w_{t+1} = w_t - \dfrac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$
- As t grows (more and more epochs) the scaling of the gradient update can become very aggressive, meaning that updates basically stop!

Adadelta

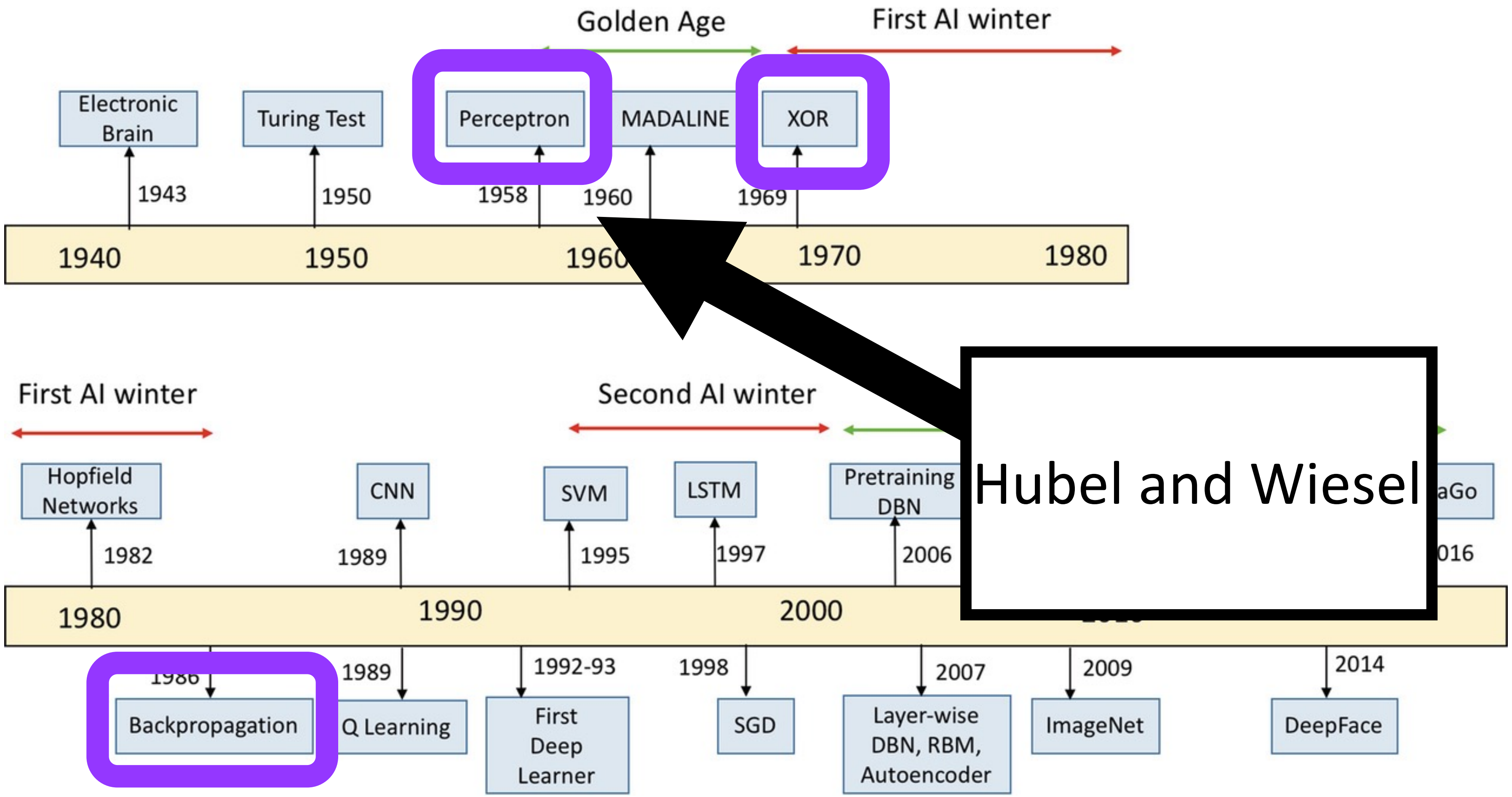
- Adagrad scales the learning rate by a factor of *all* past updates
- Solution: scale only by a *window* of past updates
  - Avoid storing all past updates in the window by applying a multiplier $g < 1$
  - Updates t time steps away will be decayed by g^t, driving down the contribution of updates that were far in the past.

ADAM

- A mix of momentum and adadelta

Second-order methods

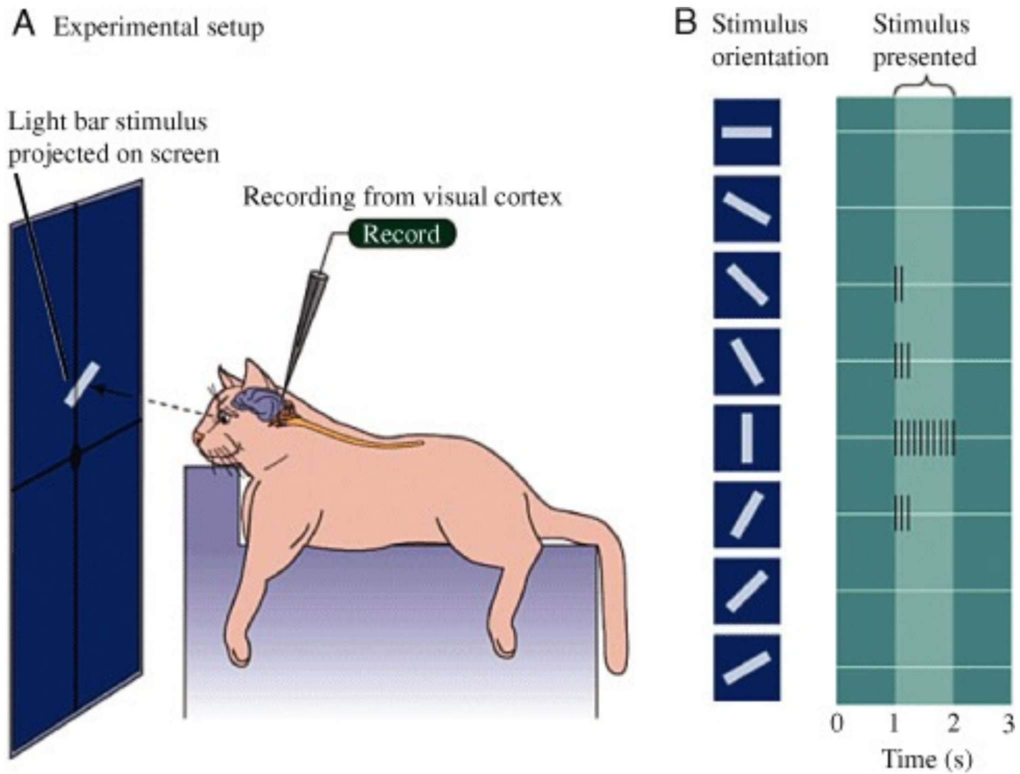
- Not often used because
  - Requires computation and storage of all 2nd order derivatives (M params -> M^2 2nd order derivatives)
  - Approximated with full dataset (which are typically v. large in DL)
- Some methods to approximate this with less overhead
  - L-BFGS

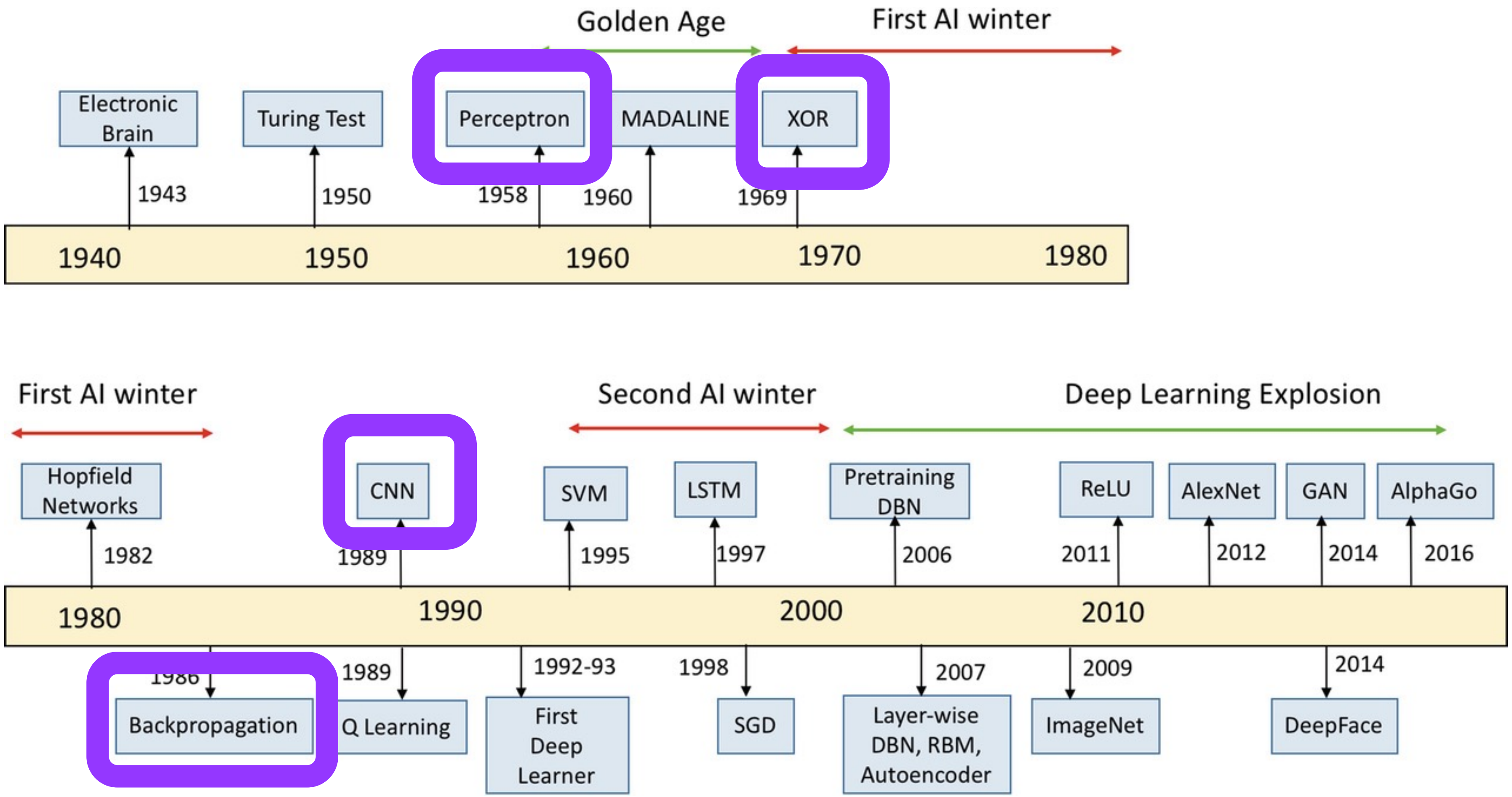


Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

An Important Discovery

- In this video, the static noise you hear is a representation of the neurons firing in response to the visual stimulus
  - https://www.youtube.com/watch?v=jw6nBWo21Zk





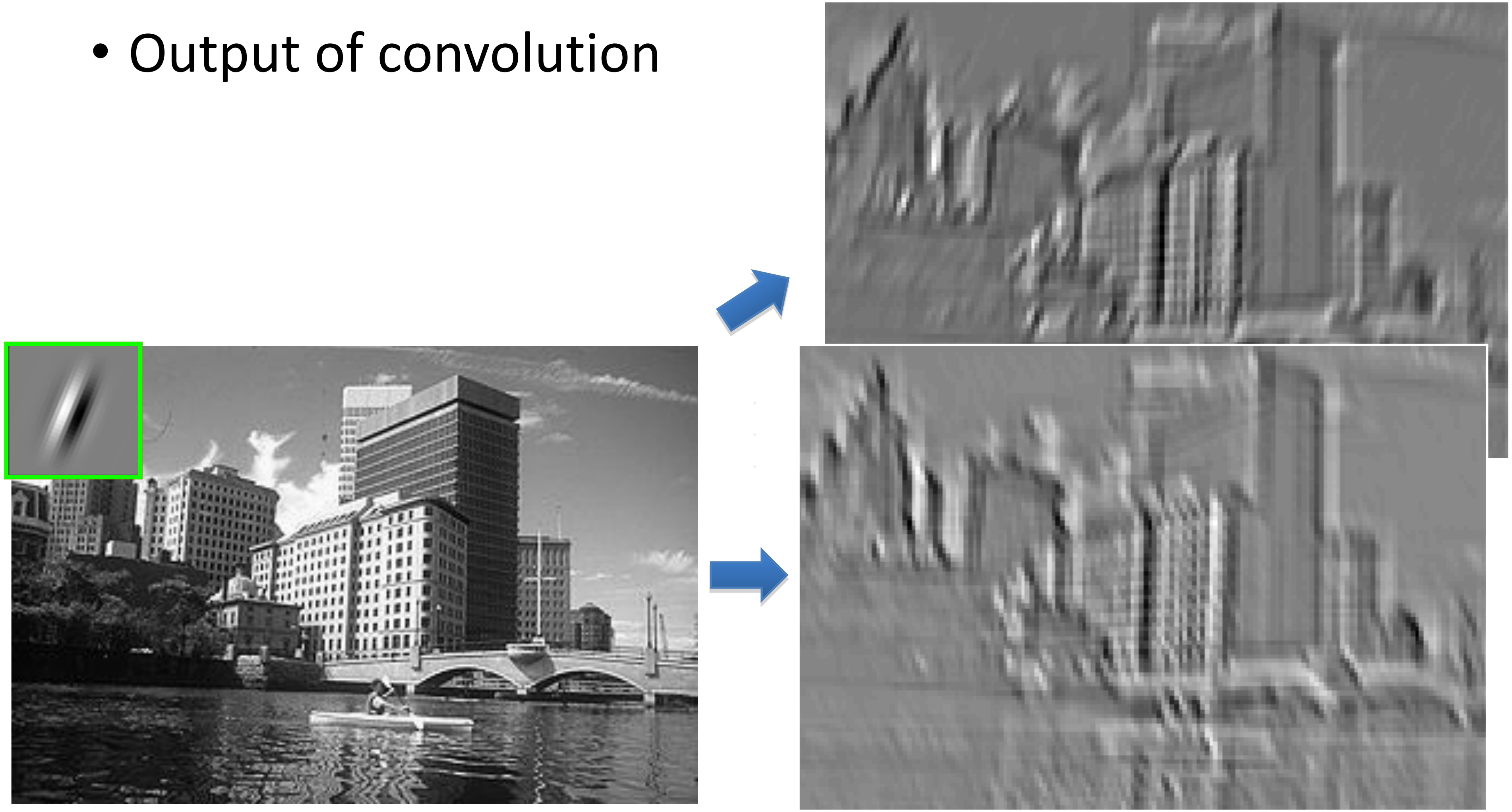
Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

CNNs: Convolution

- Hubel and Wiesel inspire the idea of convolution in neural networks
  - The same edge detector behavior can be found in multiple receptive fields
- CNNs are powerful because the same "**filter**" (i.e. edge detector) is repeatedly used on all patches of the image
  - This saves parameters, makes learning more efficient

CNNs: Convolution

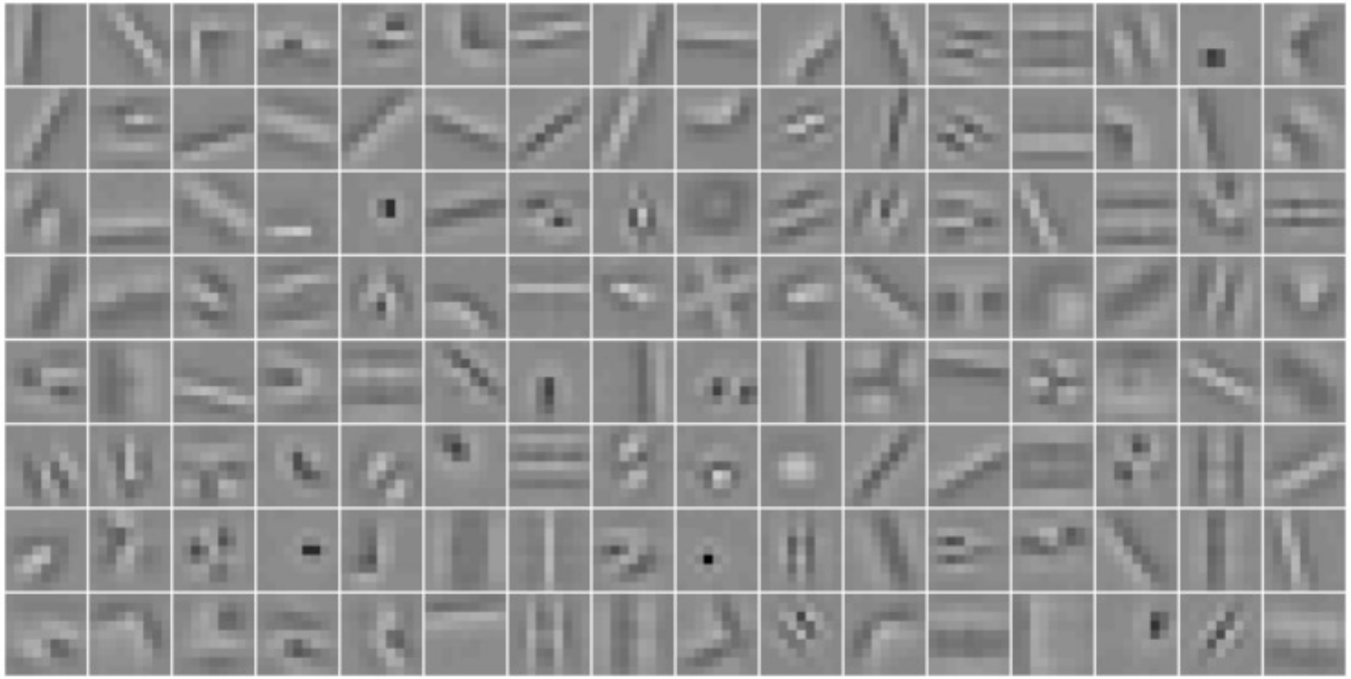
- Output of convolution



http://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/ (Fergus intro)

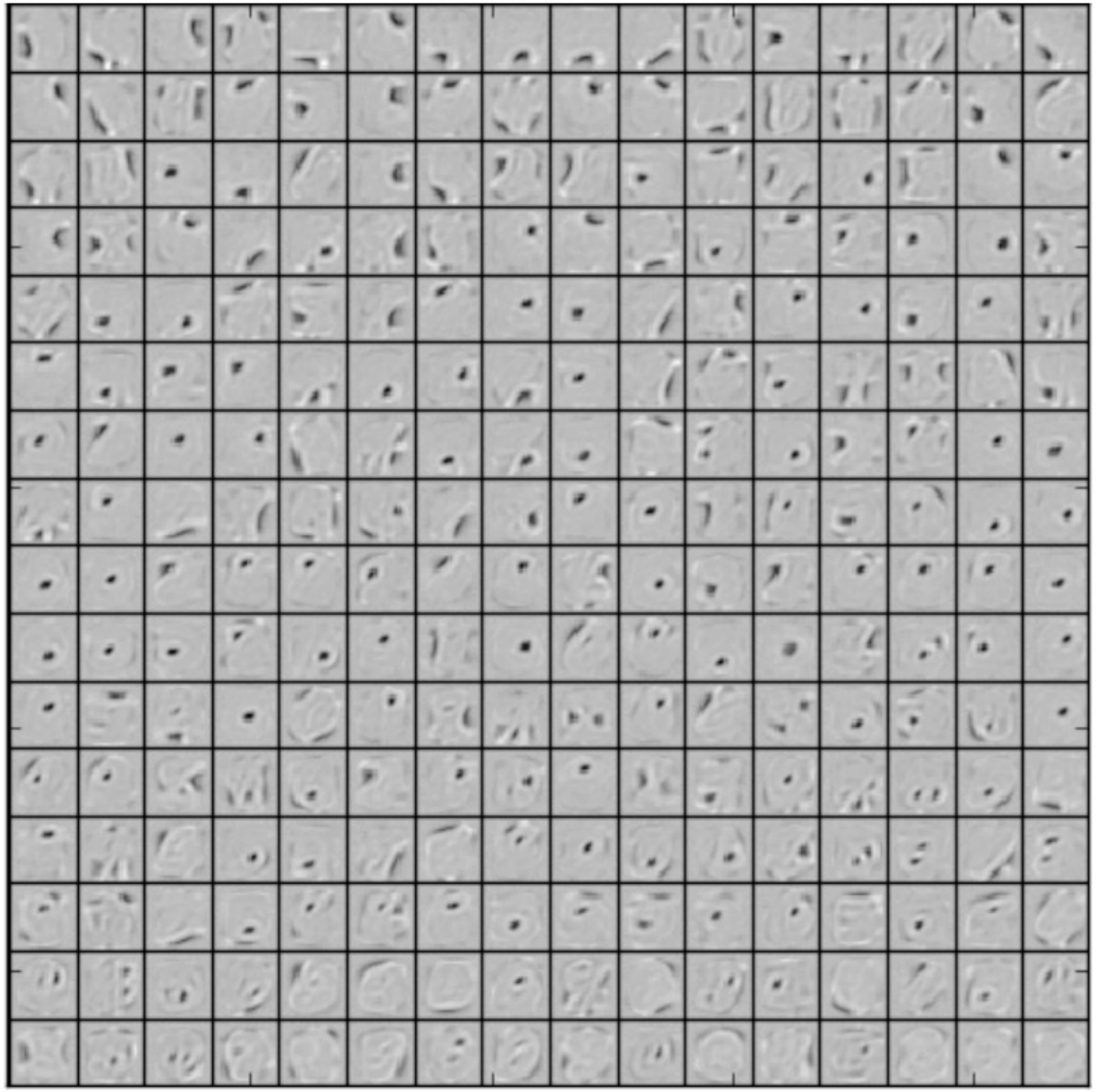
What do CNNs learn?

- When trained on images



http://yann.lecun.com/exdb/publis/pdf/lecun-eccv-12.pdf

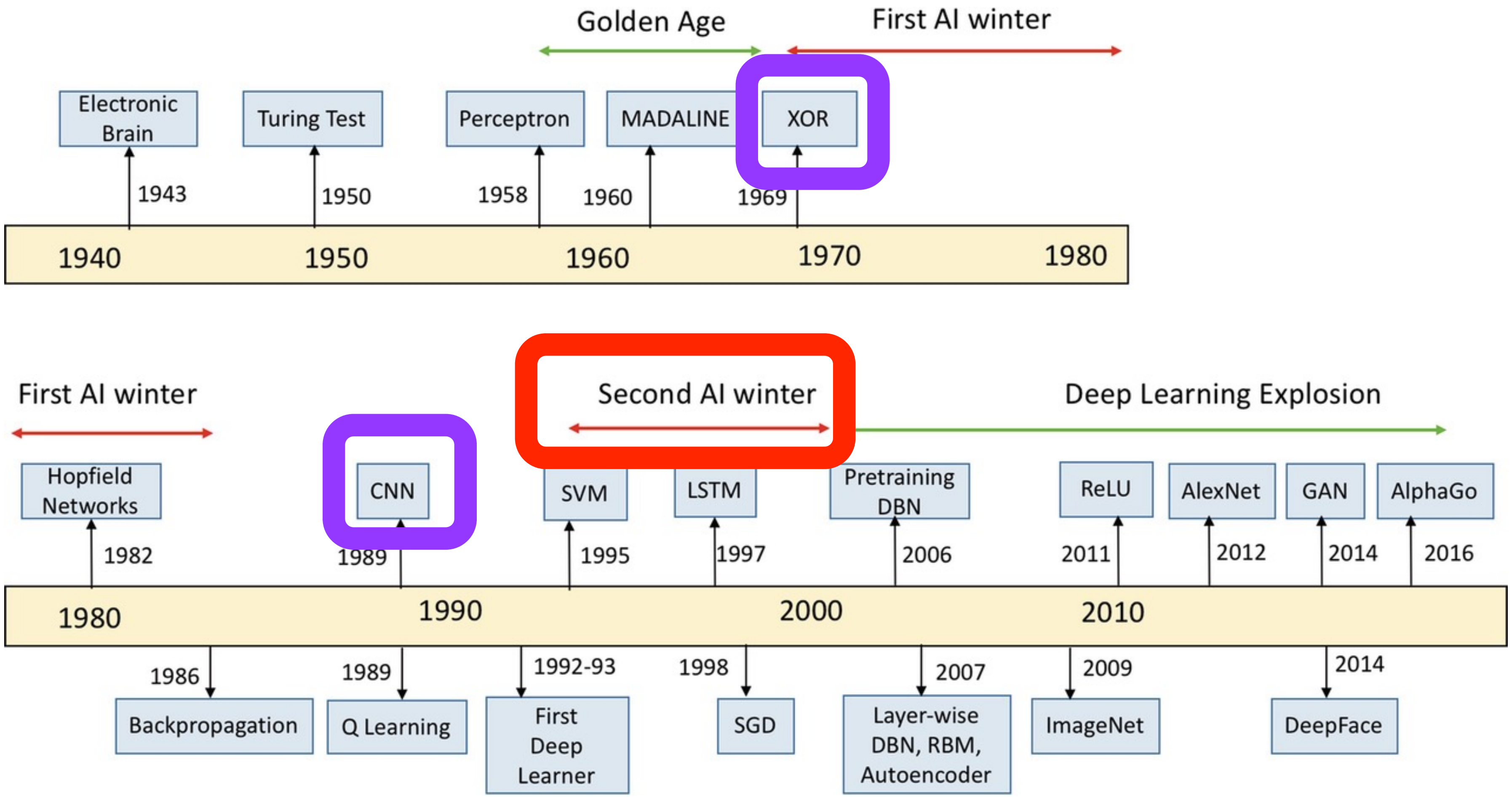
What do CNNs Learn?



https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf

CNNs

- CNNs were extremely useful for simple tasks
    - E.g. character recognition for hand written digits
- But, CNNs couldn’t handle more complex problems
    - There wasn’t enough data
    - Computers weren’t powerful enough



Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

The Second AI Winter

- AI Hype grew and grew
- Expert systems became very popular
  - Used databases of knowledge to mimic human decision making
- Companies stated “We’ve built a better brain” and declared that “[I]t is now possible to program human knowledge and experience into a computer … Artificial intelligence has finally come of age.”

https://towardsdatascience.com/history-of-the-second-ai-winter-406f18789d45

The Second AI Winter

- People became skeptical
- [John McCarthy] described the expert system MYCIN built to assist physicians.
  - He then laid out a situation where a patient has *Cholerae Vibrio* in his intestines.
  - When asked, the systems prescribed two weeks of tetracycline.
  - This would most likely kill off all the bacteria, but by then the patient would already be *dead*.

https://towardsdatascience.com/history-of-the-second-ai-winter-406f18789d45

The Second AI Winter

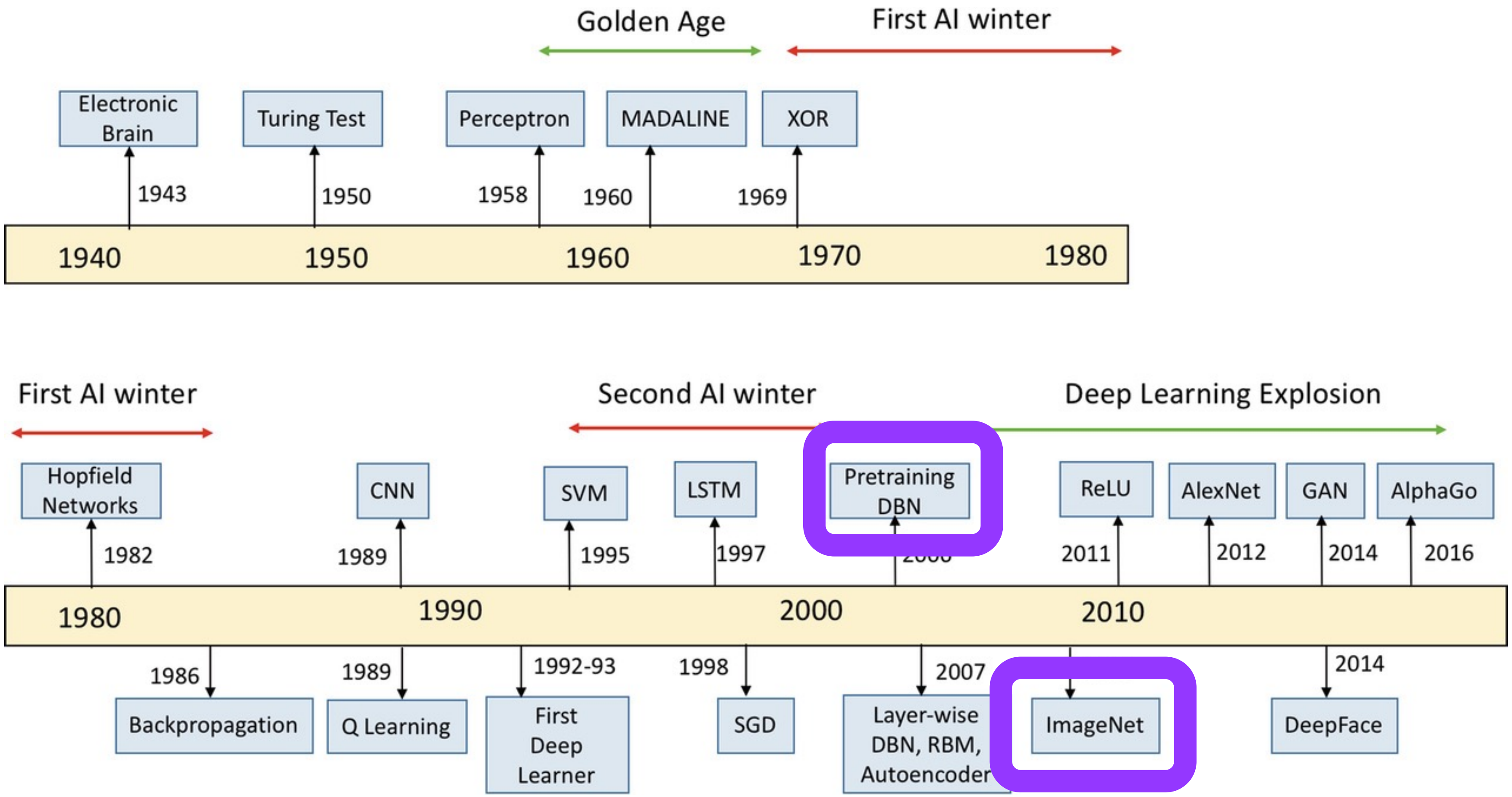
- The databases of "human knowledge" in expert systems had to be created manually
  - Rules to operate over these databases also had to be manually made
- Many tasks are too complicated for engineers to **design rules** for manually.
  - E.g. Systems for vision, medical diagnostics, etc

https://towardsdatascience.com/history-of-the-second-ai-winter-406f18789d45

The Second AI Winter

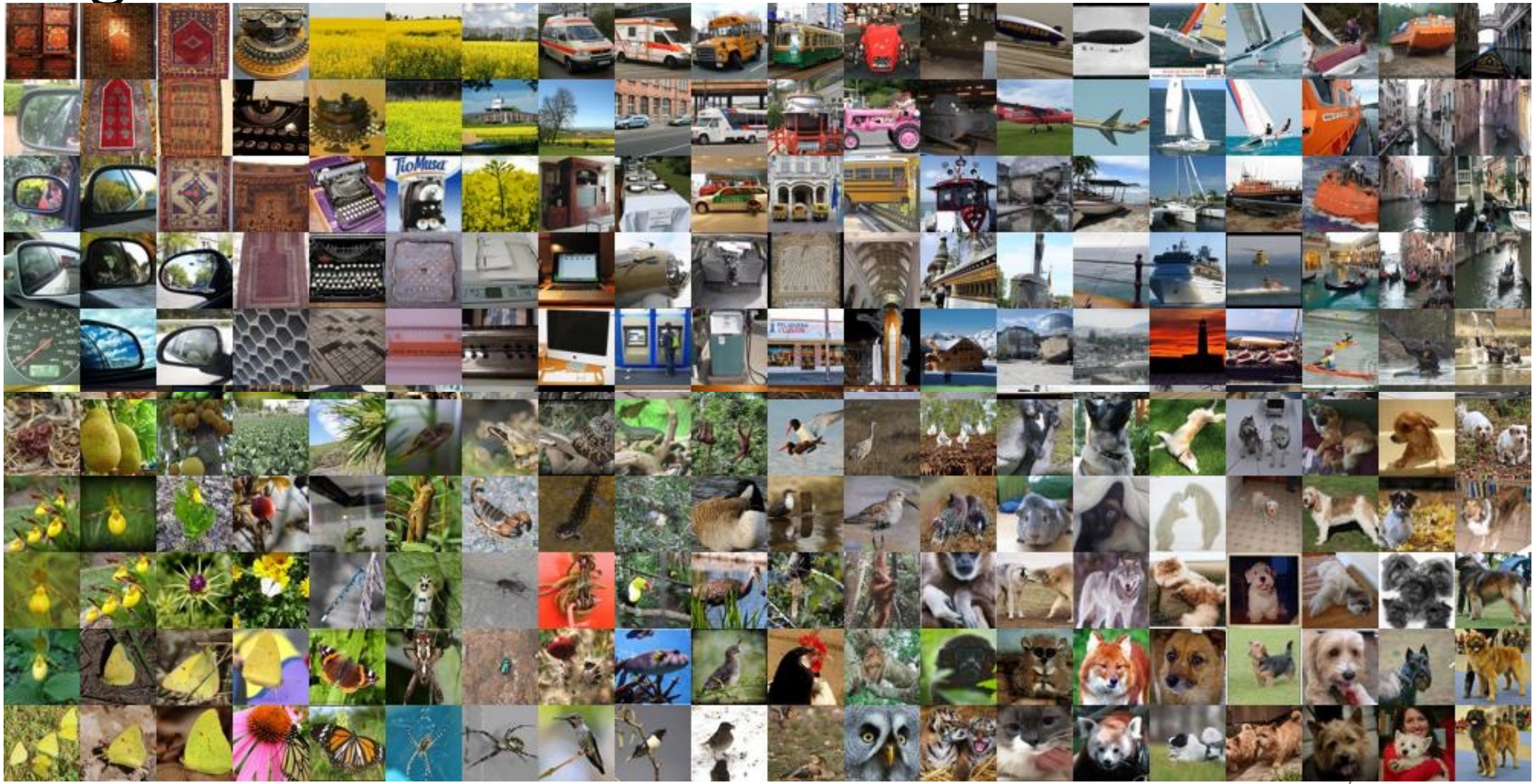
- The general interest in AI declined as the expectations could not be met.
- Many AI companies closed their doors.
- The AAAI conference that attracted over 6000 visitors in 1986 quickly decreased to just 2000 by 1991.

https://towardsdatascience.com/history-of-the-second-ai-winter-406f18789d45



Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

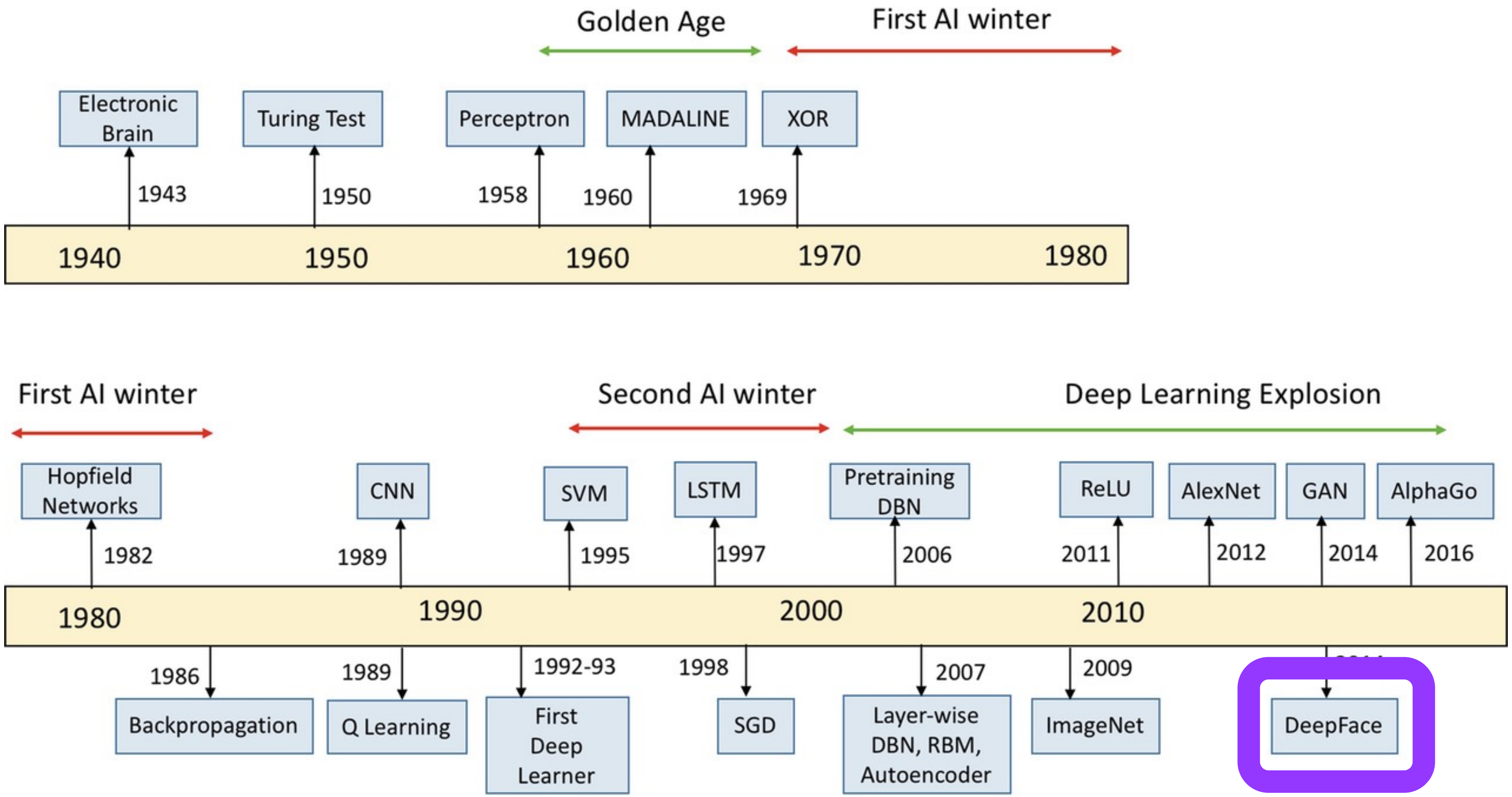
ImageNet



http://cs.stanford.edu/people/karpathy/cnnembed/

ImageNet Enables Learning

- The size of ImageNet, and the increasing speed of computers, allows for CNNs to become world class object detectors!
- In contrast to expert systems, CNNs learn their database of filters, and the functions (rules) that operate over them
  - Much more powerful
  - Generalize well to novel images
  - Generalize well to new problem domains

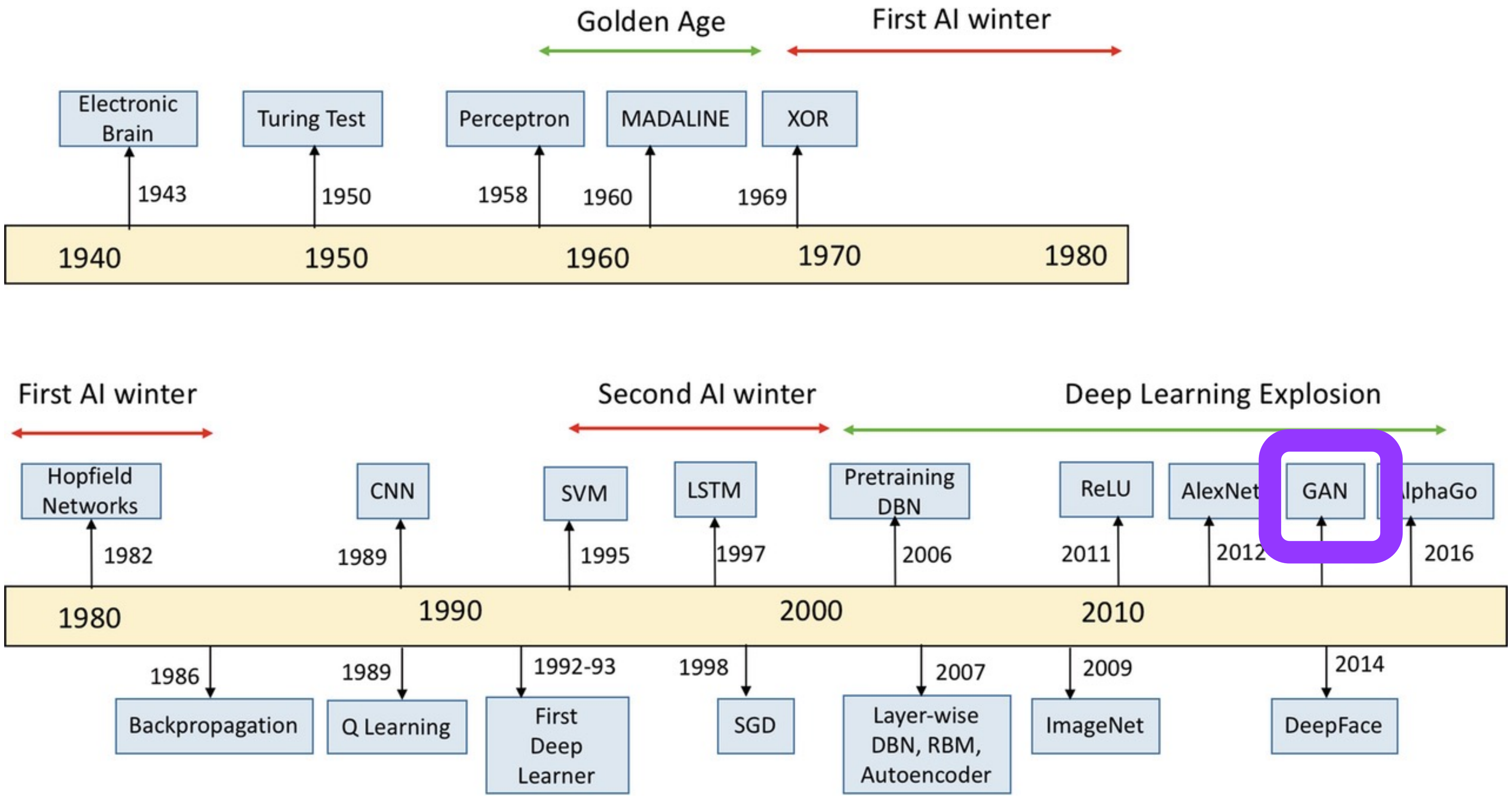


Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

DeepFace

- Facebook used images **uploaded and tagged by its users** to build a face recognition system with 97.3% accuracy
- At the time, was the largest facial dataset to-date, an identity labeled dataset of four million facial images belonging to more than 4,000 identities.

https://research.fb.com/publications/deepface-closing-the-gap-to-human-level-performance-in-face-verification/

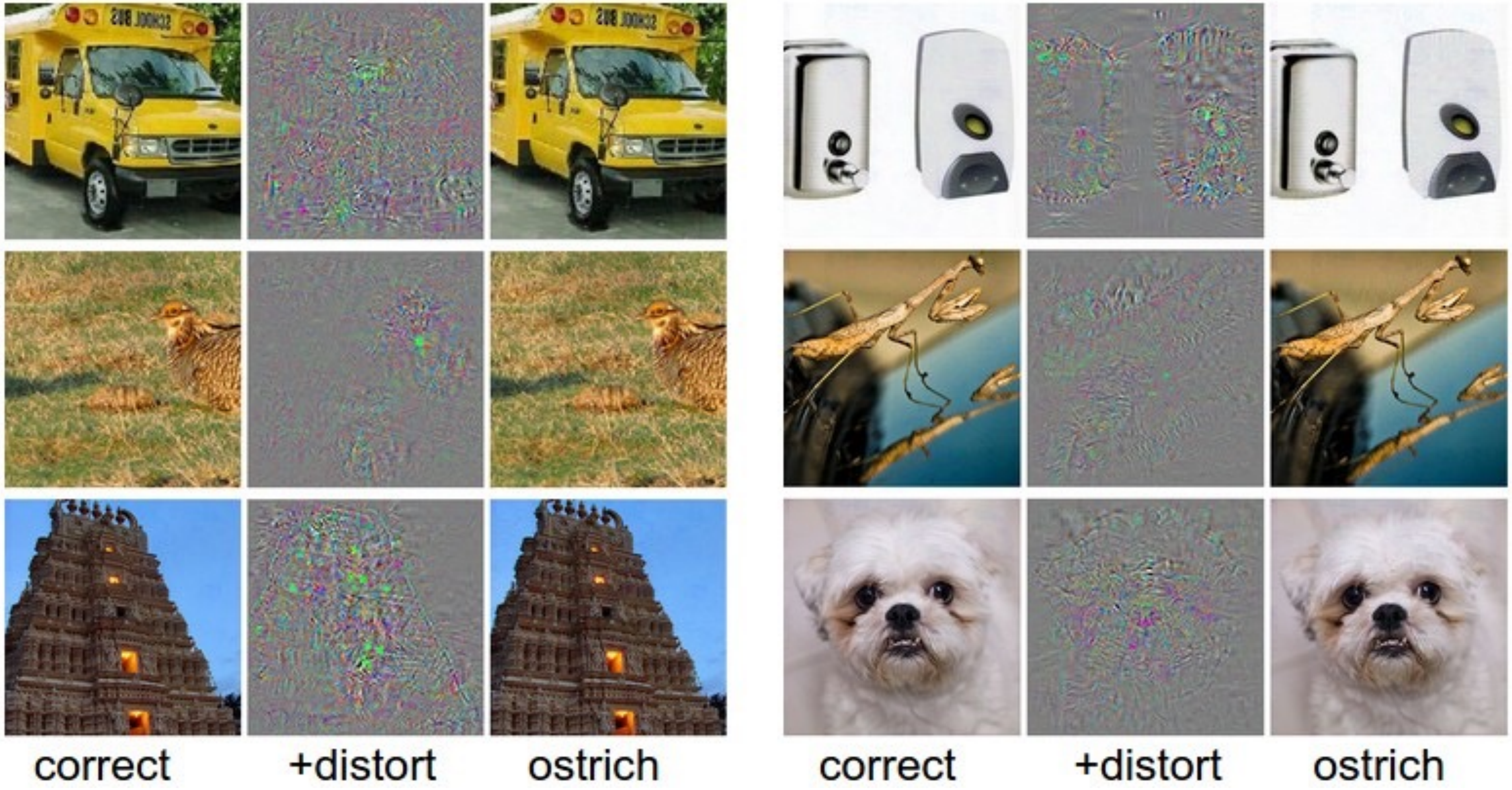


Kamath U., Liu J., Whitaker J. (2019) Introduction. In: Deep Learning for NLP and Speech Recognition. Springer, Cham

First: Adversarial Examples

Adversarial Examples

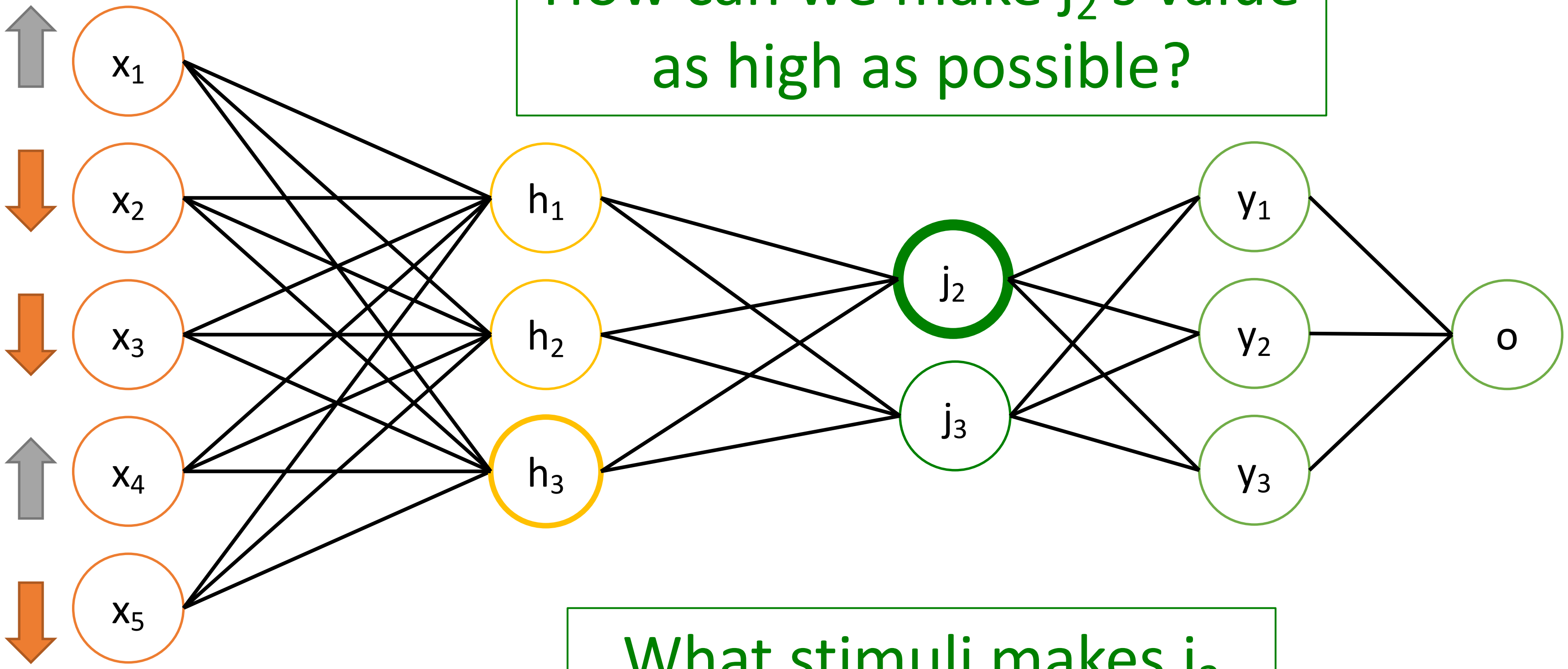
- We can make small changes to an image (nearly imperceptible!) and cause a network to misclassify
- Extreme implications for e.g. self driving cars



http://karpathy.github.io/2015/03/30/breaking-convnets/

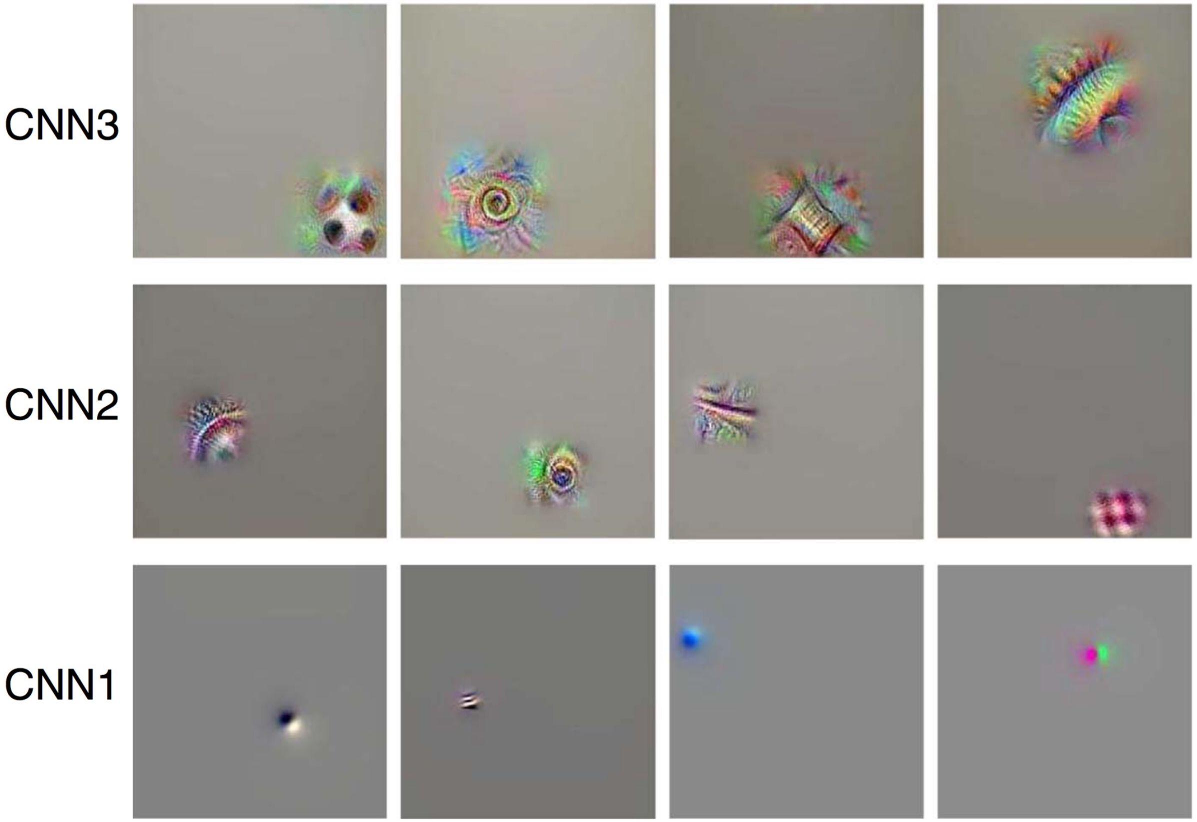
How are adversarial examples made?

Recall:



How can we make j_2's value as high as possible?

What stimuli makes j_2 fire maximally?



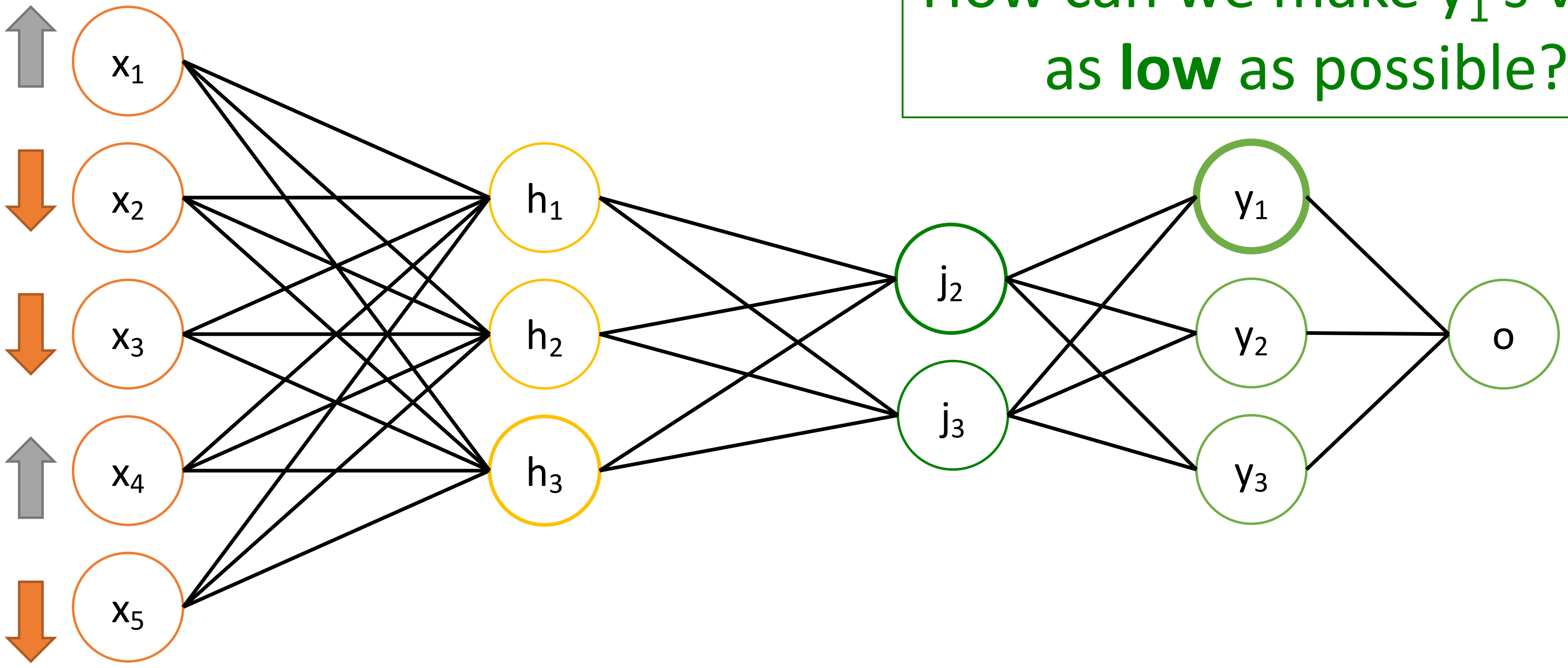
Horikawa & Kamitani (2017)

How are adversarial examples made?

- An adversarial example forces the model to predict the wrong class
  - Sometimes a specific wrong class

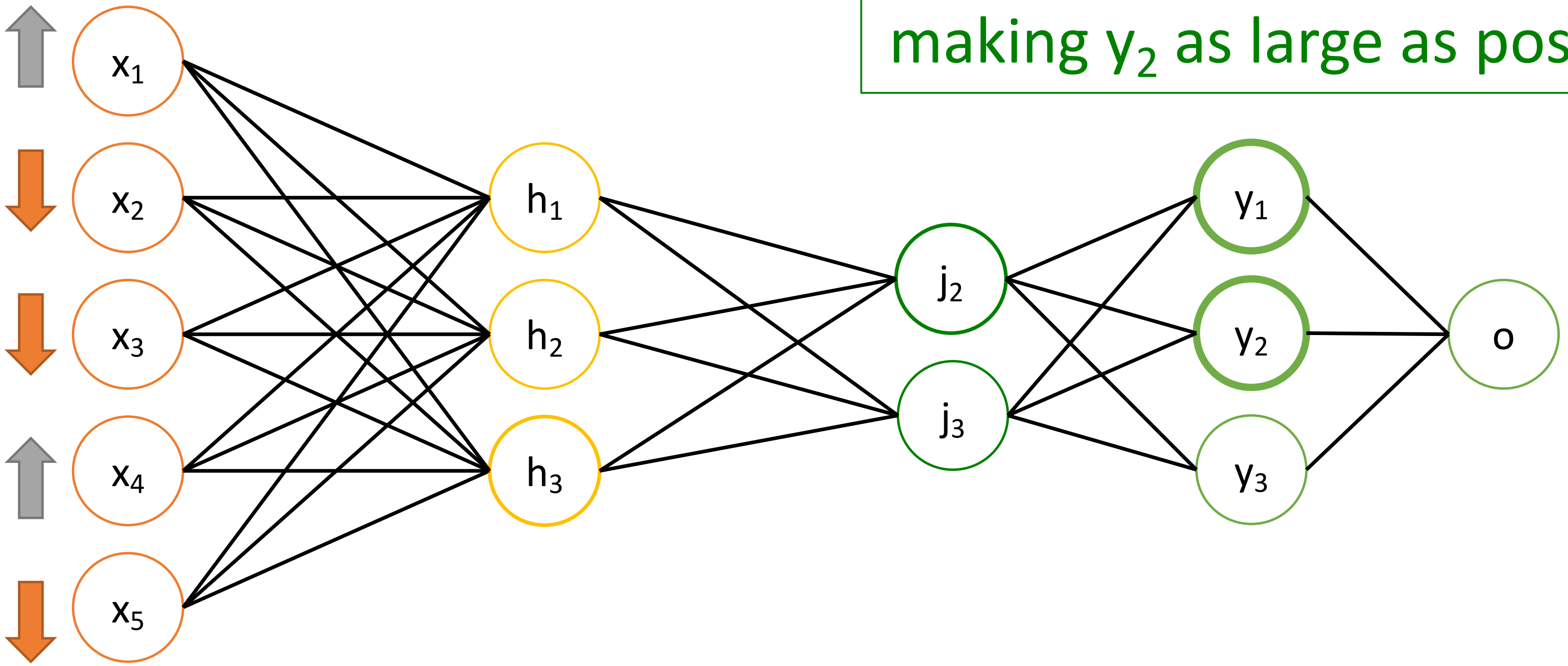
Correct class is y_1

How can we make y_1's value as **low** as possible?



Correct class is y_1

How can we make y_1's value as **low** as possible while also making y_2 as large as possible?



Removing Stop Signs with Stickers

https://www.usenix.org/system/files/conference/woot18/woot18-paper-eykholt.pdf

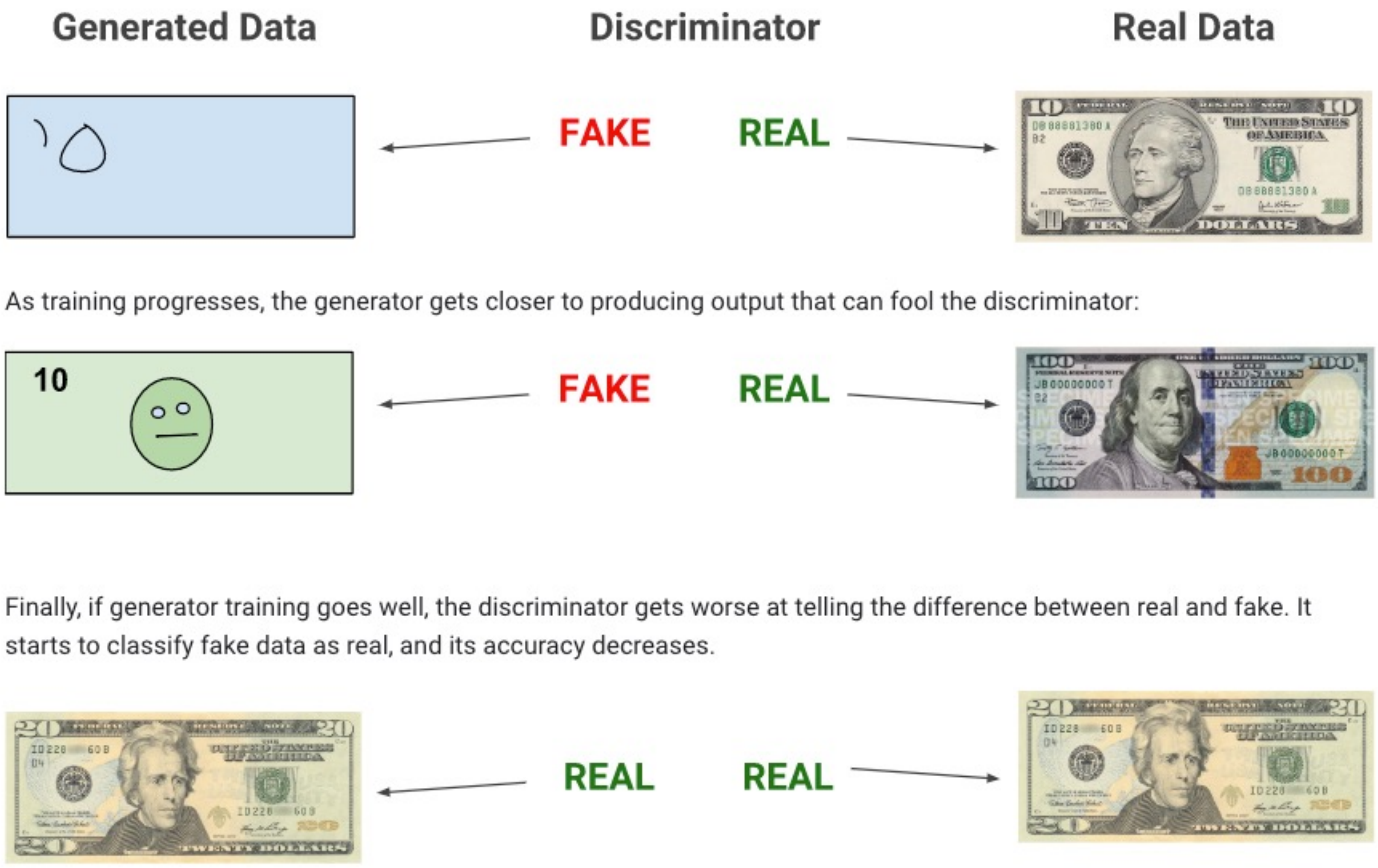
Making Stop Signs with Stickers

Making Stop Signs with Stickers

GAN

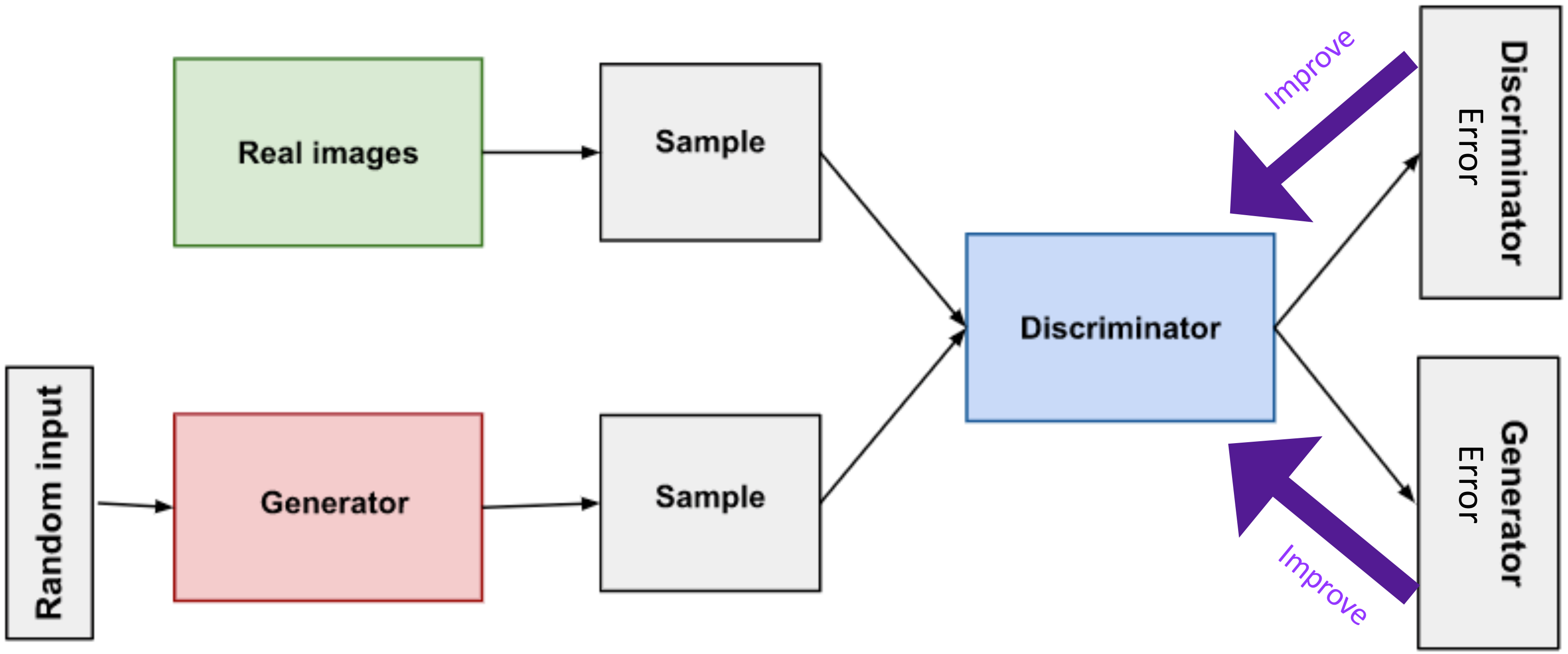
- Generative Adversarial Network
- Two dueling neural networks
  - One trained to generate images
  - One trained to distinguish generated images from true images

GAN



https://developers.google.com/machine-learning/gan/gan_structure

GAN



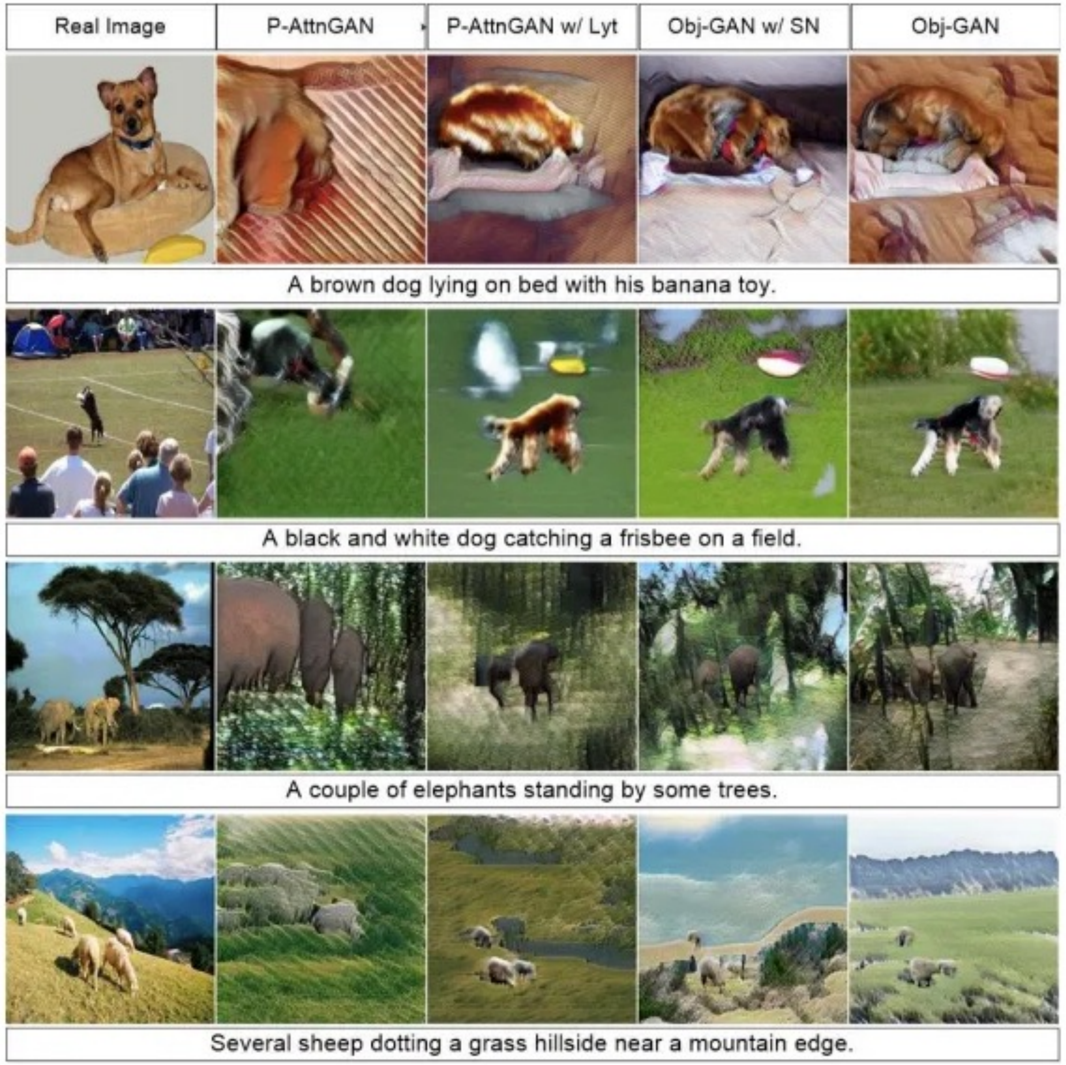
https://developers.google.com/machine-learning/gan/gan_structure

GAN

Figure 1: Class-conditional samples generated by our model.

https://medium.com/syncedreview/biggan-a-new-state-of-the-art-in-image-synthesis-cf2ec5694024

GAN



https://syncedreview.com/2019/06/25/microsoft-obj-gan-turns-words-into-complex-scenes/

GAN

https://medium.com/syncedreview/gan-2-0-nvidias-hyperrealistic-face-generator-e3439d33ebaf

DeepFace or DeepFake?

- https://www.creativebloq.com/features/deepfake-examples

Resources

- Programming resources for training your own NNs
  - Tensorflow https://www.tensorflow.org/
  - Keras https://keras.io/
  - Pytorch https://pytorch.org/
  - For intuition: http://playground.tensorflow.org/
- Short course on deep learning (Nando De Freitas)
  - https://www.youtube.com/playlist?list=PLjK8ddCbDMphIMSXn-w1IjyYpHU3DaUYw
- Commentary on AlphaGo
  - https://www.youtube.com/watch?v=UMm0XaCFTJQ
  - https://www.youtube.com/watch?v=g-dKXOlsf98
- Other fun videos
  - Geoff Hinton is in this one! Neural Net stuff is towards the end
    - https://www.youtube.com/watch?v=yxxRAHVtafI
  - Fei Fei Li's Ted Talk (Creator of ImageNet)
    - https://www.ted.com/talks/fei_fei_li_how_we_re_teaching_computers_to_understand_pictures?language=en