Deep Learning on Azure

Deep Learning Crash Course

Jarek Kazmierczak
MTC Silicon Valley

Agenda

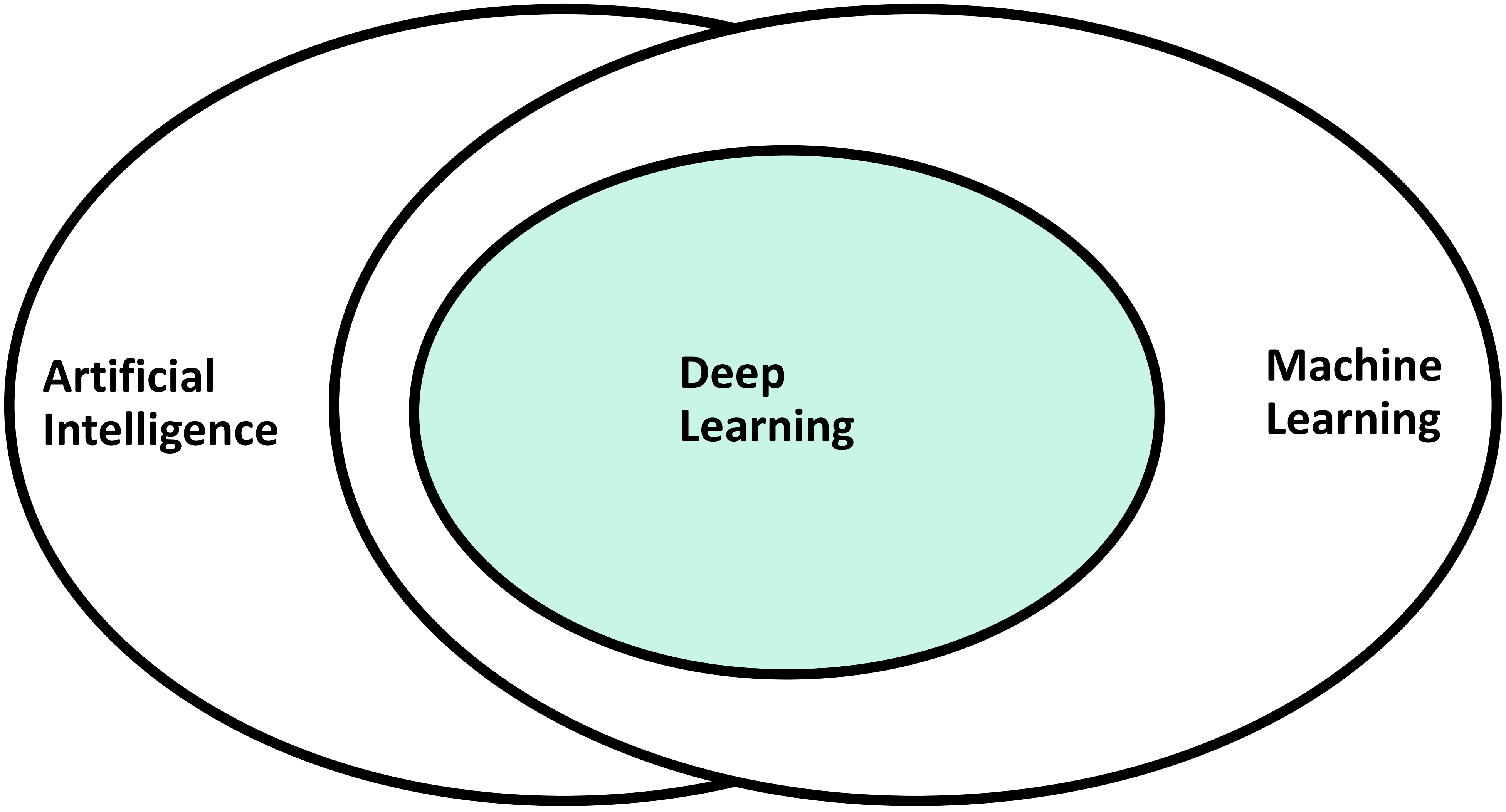
Morning session:

- Introductions
- Deep Learning Crash Course
- Overview of Microsoft Cognitive Toolkit
- Hands-on labs:
  - Lab 1 - Multiclass Logistic Regression – Computer Vision

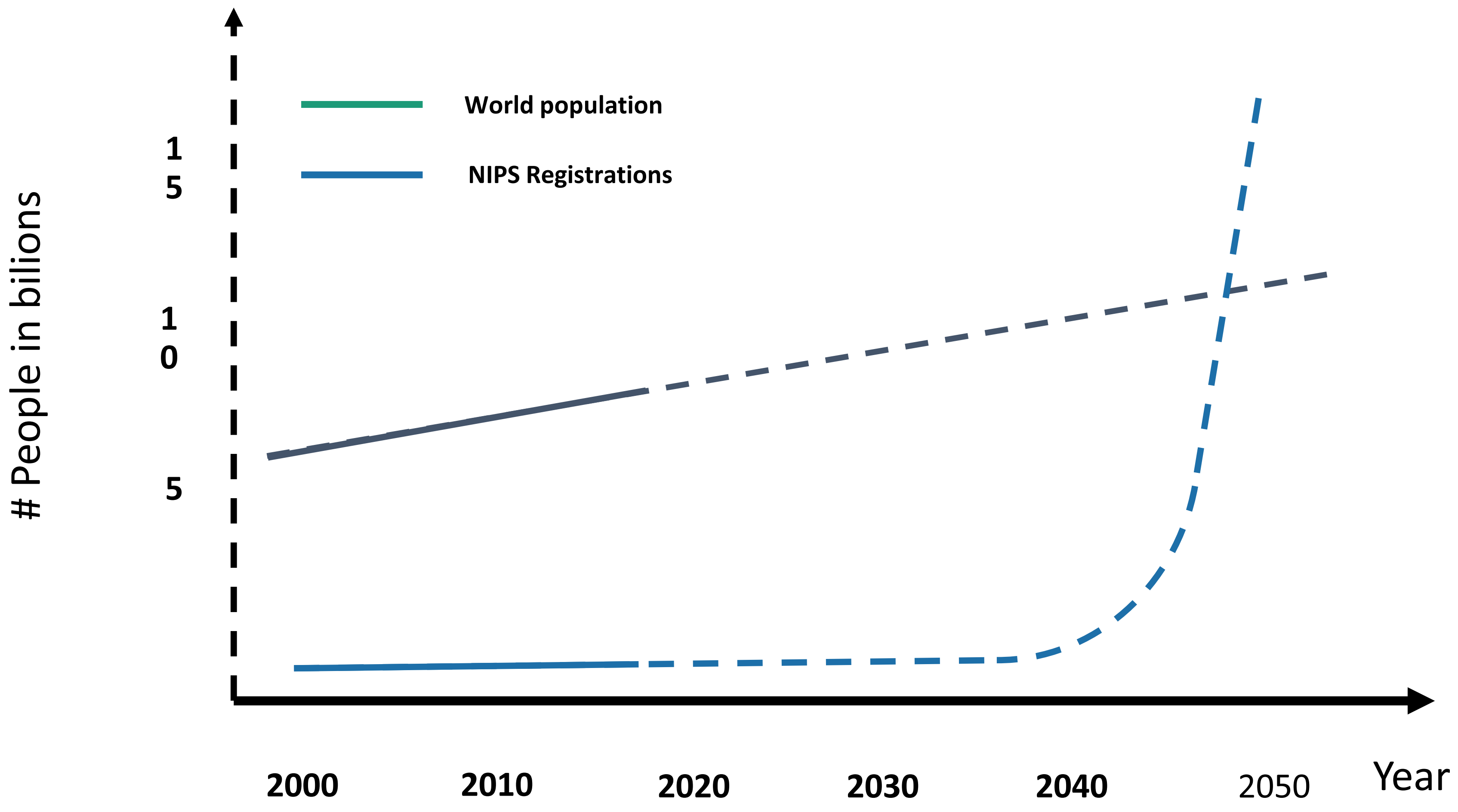
Afternoon session:

- Hands-on labs:
  - Lab 2 - Fully Connected Neural Network – Computer Vision
  - Lab 3 – LSTM – Time series with IoT data

Artificial Intelligence, Machine Learning and Deep Learning



Deep Learning Hype



Have we been there before?

2006

1940s 1960s 1980s 2000s 2020s

Early Enthusiasm

Dose of Reality

Return of Neural Nets

Rise of Deep Learning

Source: Stuart J. Russel, Peter Norvig

The Deep Learning Triumv[ei]rate

LeCun: "You have to realize that deep learning .... is really a conspiracy between Geoff Hinton and myself and Yoshua Bengio"

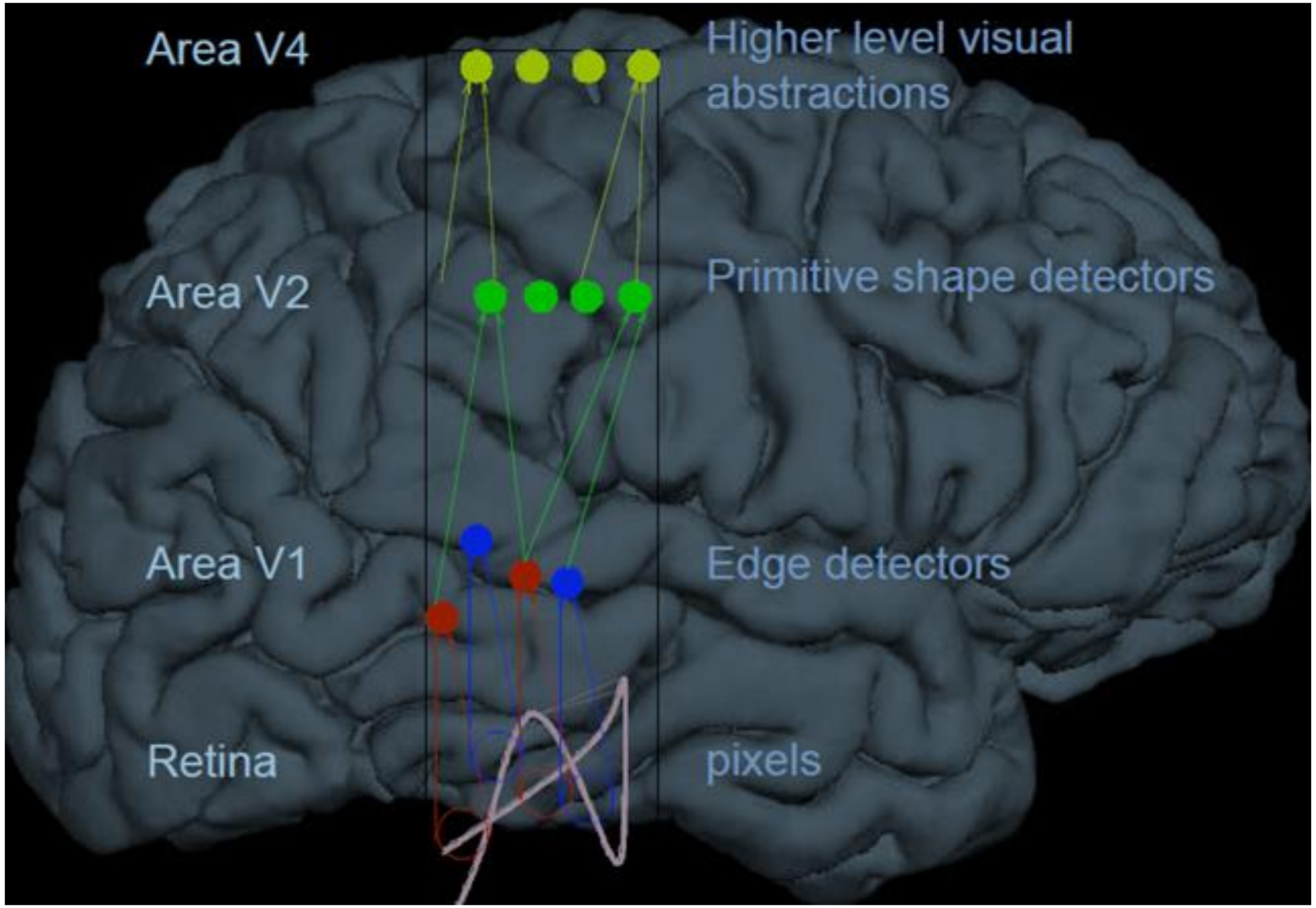
Geoff Hinton | Yann LeCun | Yoshua Bengio

Latin : Trium - {ver,vir} - ate

English : Three - {truth, men} - official

Why go deep?

- Deep learning algorithms attempt multiple levels of representation of increasing complexity/abstraction
- Brains have a deep architecture
- Deep Learning has been successful in tasks that have been a challenge for “traditional ML”

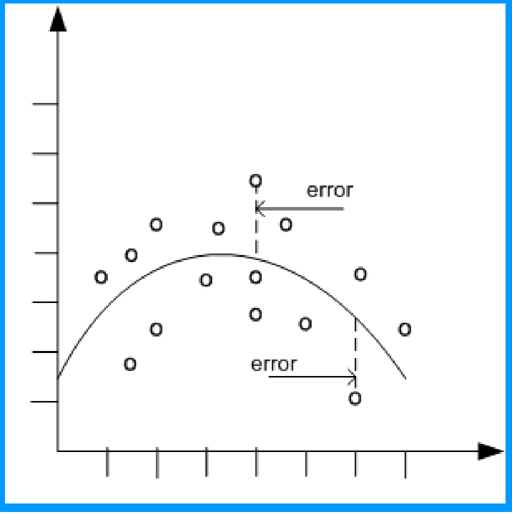


What is Machine Learning?

Tom Mitchell (1998). Well-posed Learning Problem: A computer program is said to *learn* from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

The Task, *T*

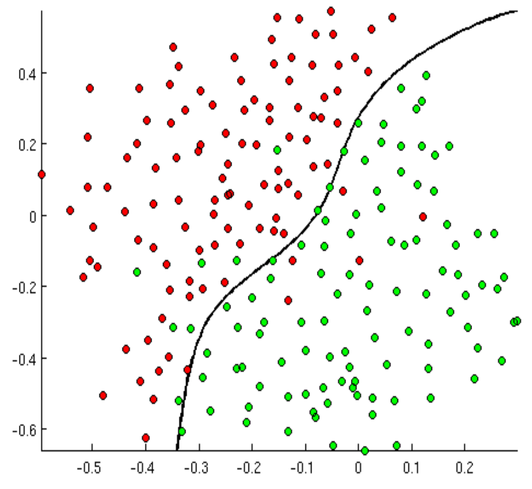
Predict a number
Regression



What will referral fee revenue be in Q3?

What will assets under management be in 2017?

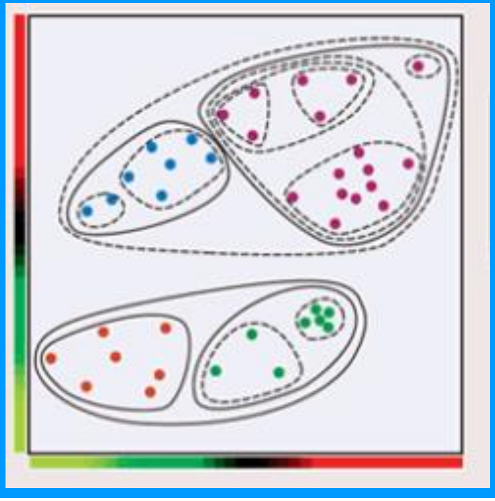
Predict a Class
Classification



What is propensity of customer to purchase a variable annuity?

Probability the customer will churn to competitor?

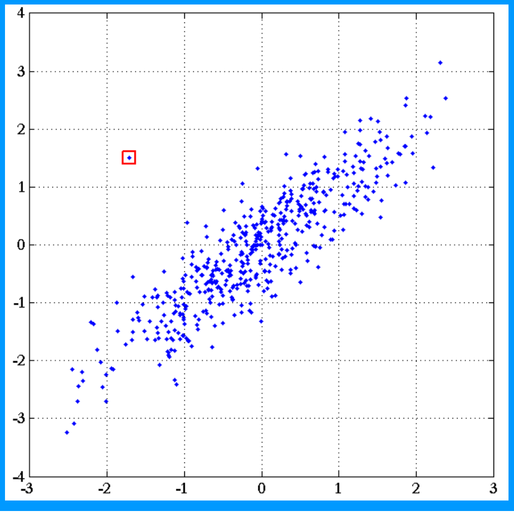
Find groups/patterns
Clustering



Find similar customers for a new “wealth builder” segment.

What is the profile of customers with 3+ products?

Find unusual items
Anomaly Detection



Identify fraudulent expense report filings.

Traditional ML Vs DL

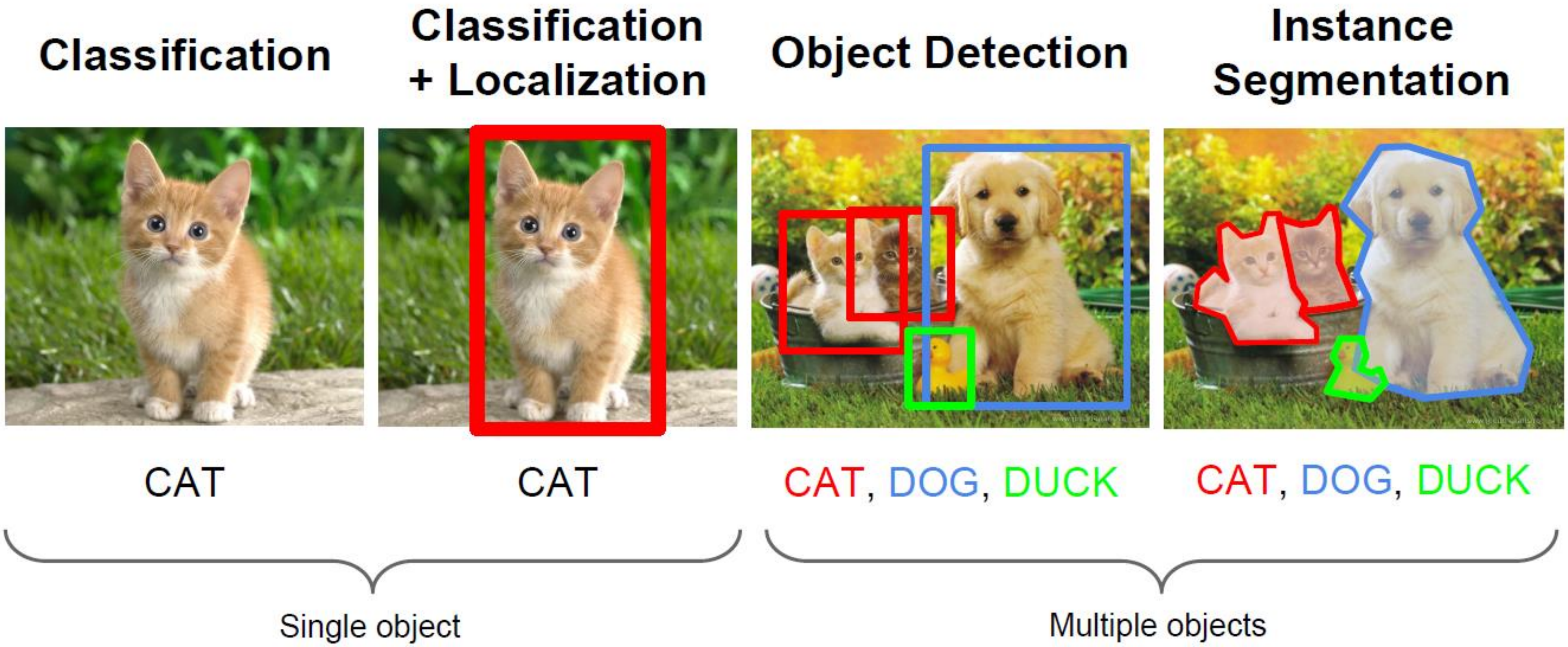
Traditional ML requires manual feature extraction/engineering

Deep learning can automatically learn features in data

Feature extraction for unstructured data is very difficult

Deep learning is largely a "black box" technique, updating learned weights at each layer

Computer vision tasks



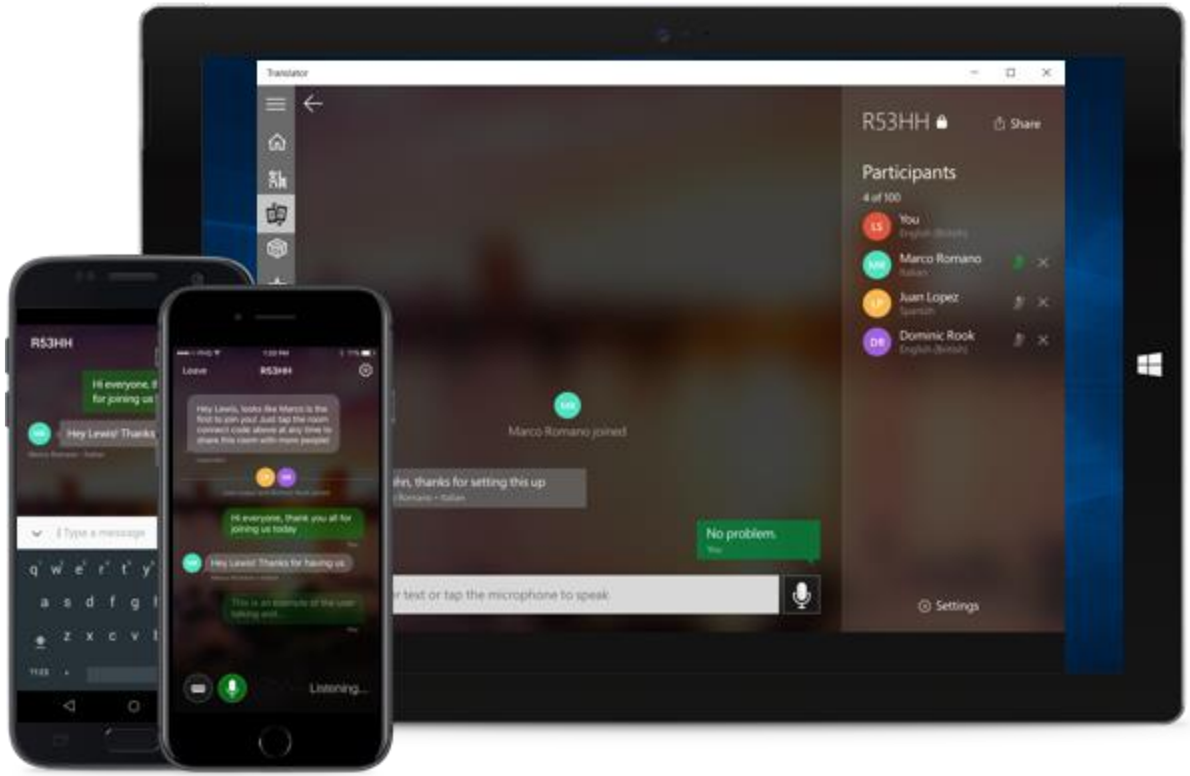
Speech recognition tasks

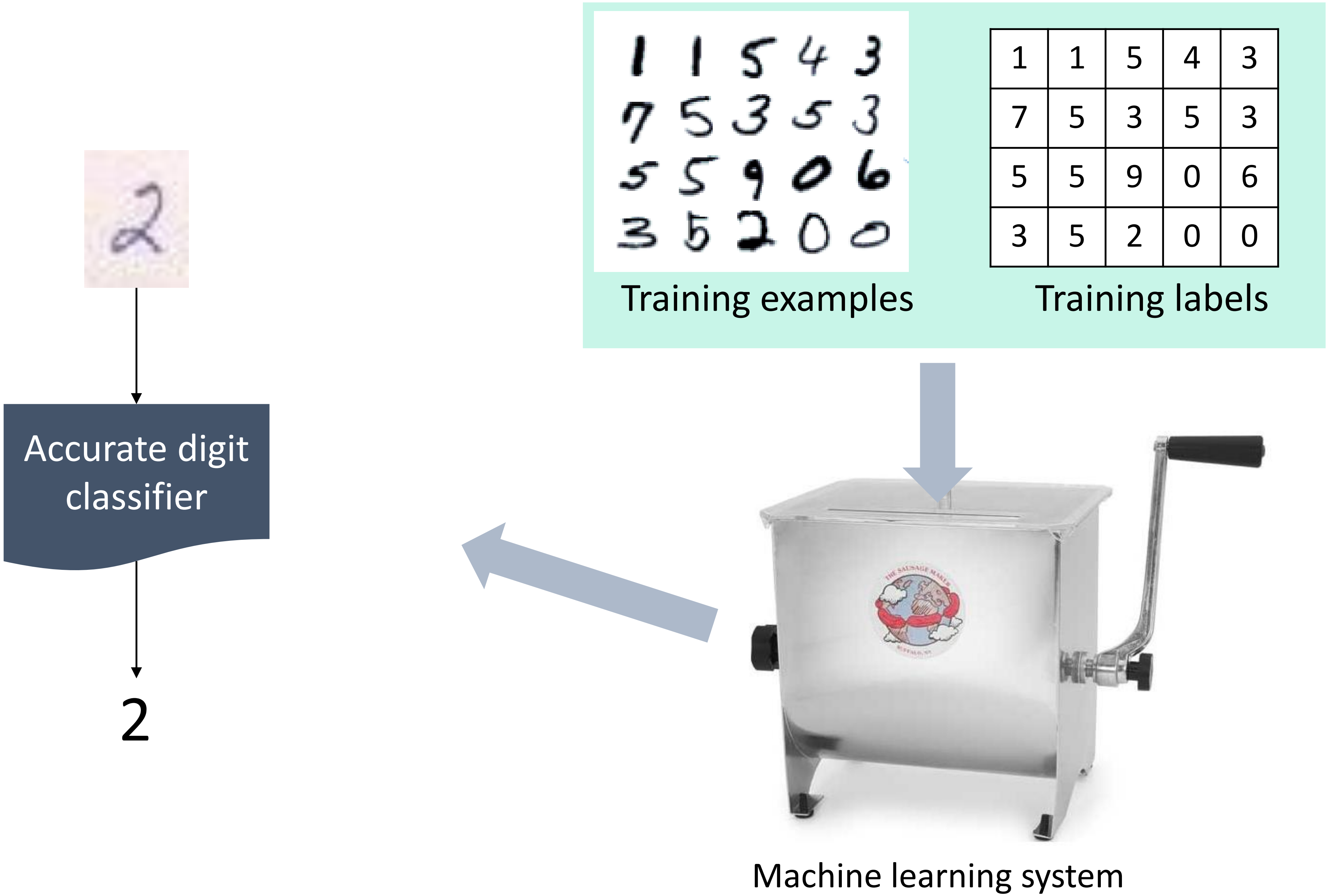
- Microsoft 2016 research system for conversational speech recognition
- 5.9% word-error rate
- enabled by CNTK's multi-server scalability

[W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, G. Zweig: "Achieving Human Parity in Conversational Speech Recognition," https://arxiv.org/abs/1610.05256]

Language translation and understanding tasks

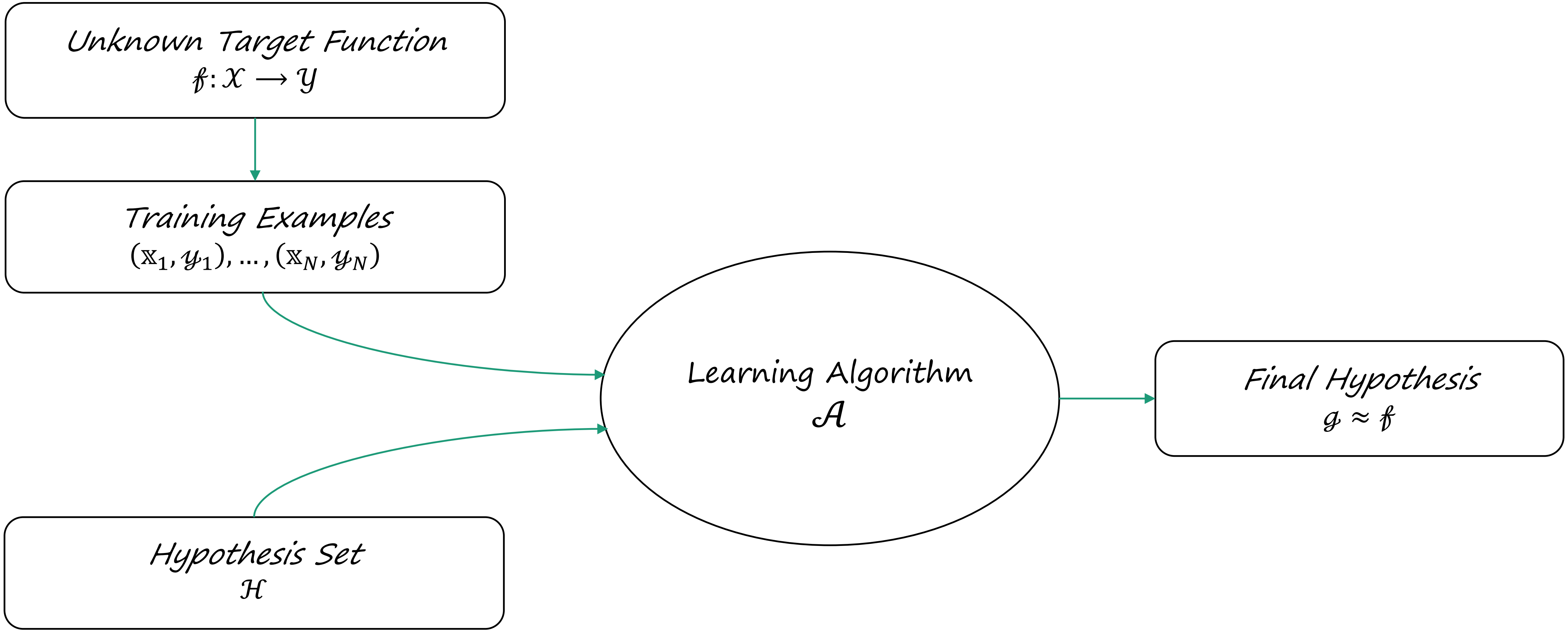
- Microsoft Translator live is an in-person, multi-device translation service for two or more participants, speech or text.
- Start a conversation, share the code and break the language barrier.
- 9 speech languages and 60 text languages.
- Apps on iOS, Android, Windows UWP and web. API to manage the conversation.



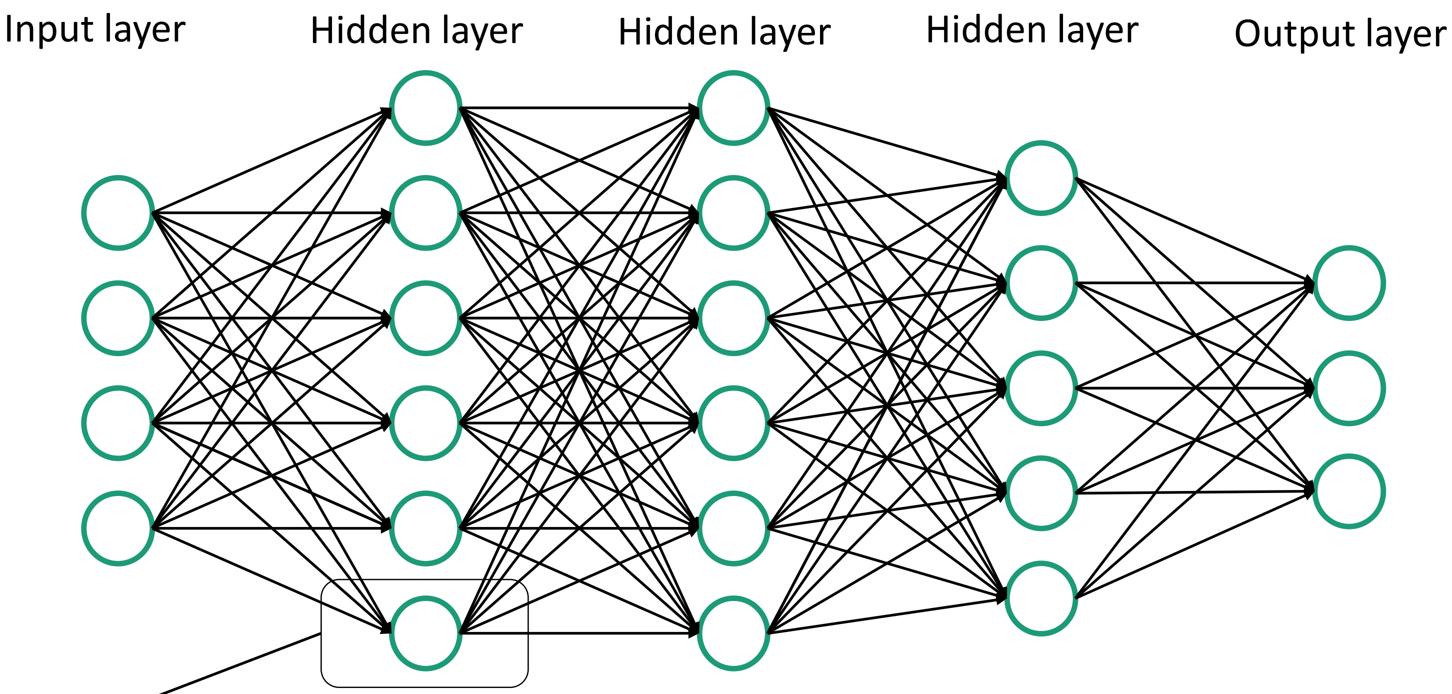


1 1 5 4 3
7 5 3 5 3
5 5 9 0 6
3 5 2 0 0
Training examples
Training labels
Accurate digit classifier
2
Machine learning system

Simplified learning model

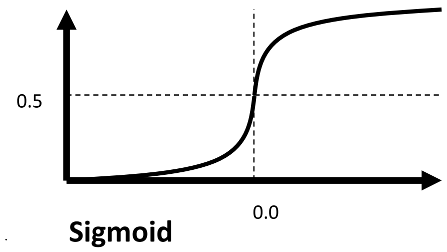


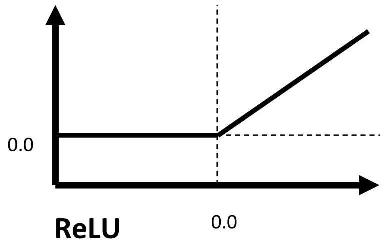
Artificial Neural Networks - ANNs



$$a_j = \sigma\left(\sum_{i=1}^{n} w_i x_i + b_j\right)$$

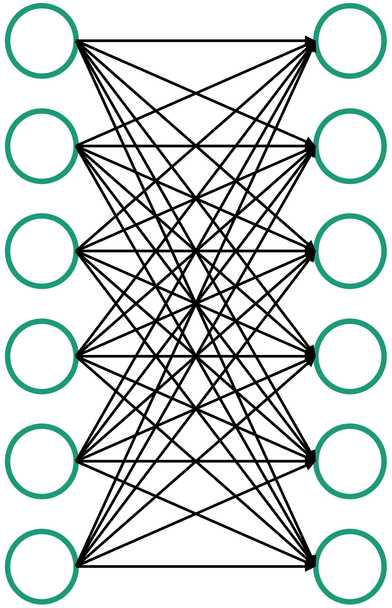
Activation functions





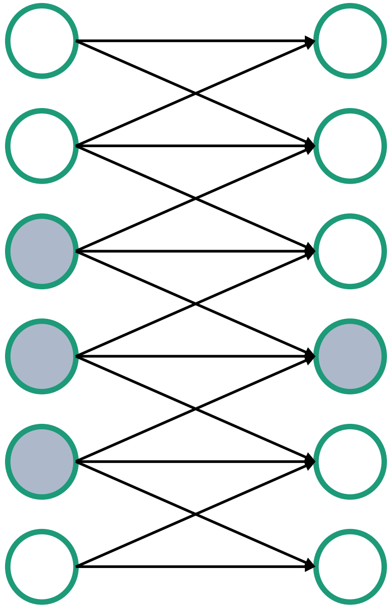
Neurons can connect in various ways …

“Dense”



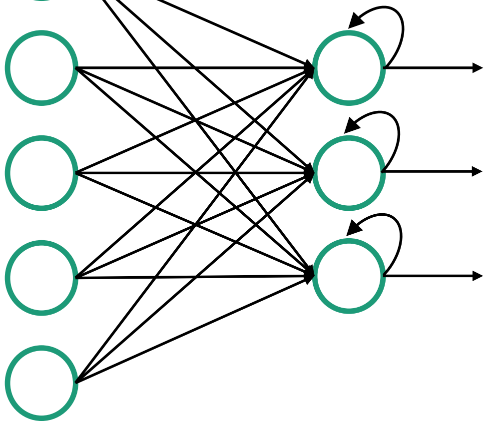
Fully **C**onnected **N**eural **N**etworks

“Sparse”



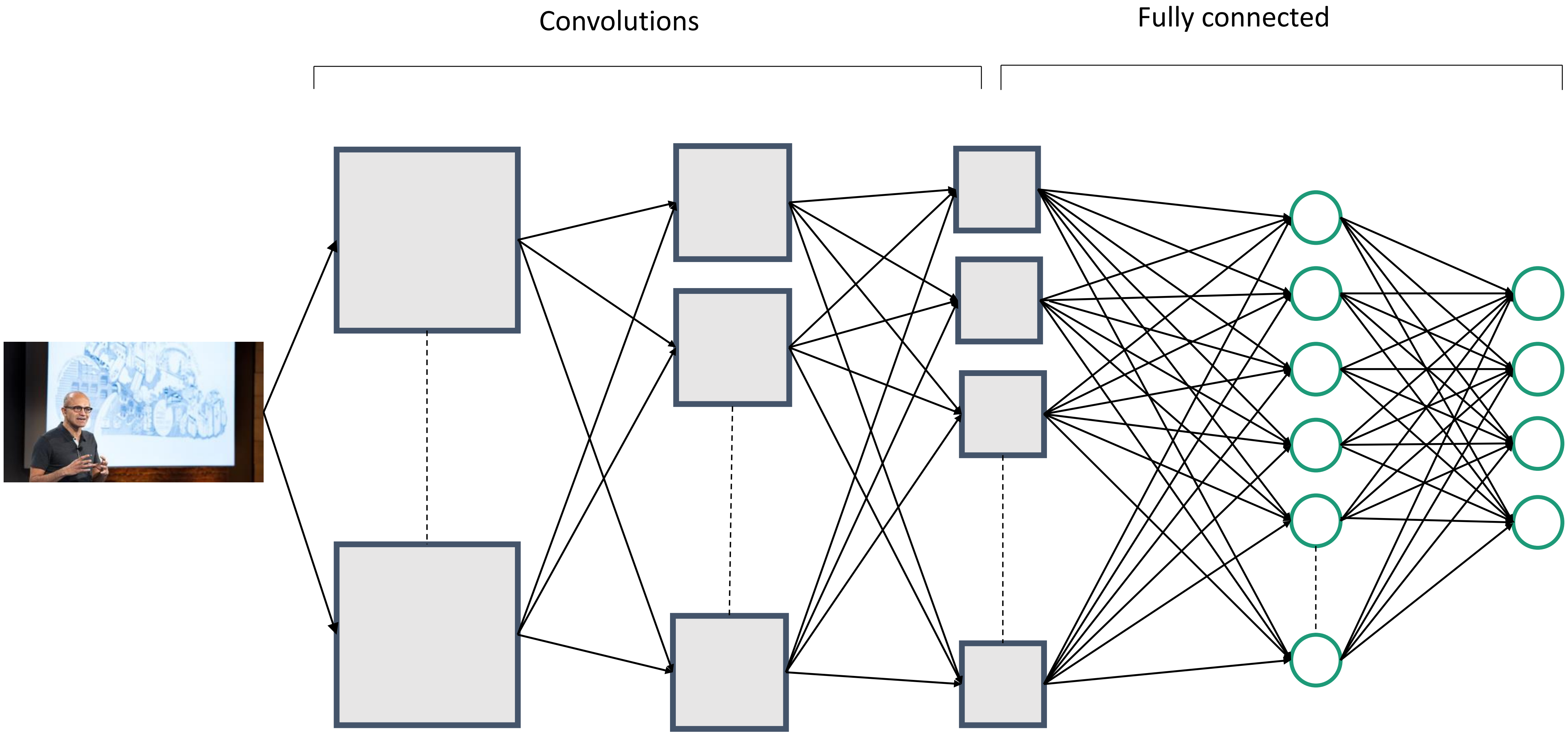
Convolutional **Neural N**etworks

“Feedback loops”



Recurrent Neural Networks

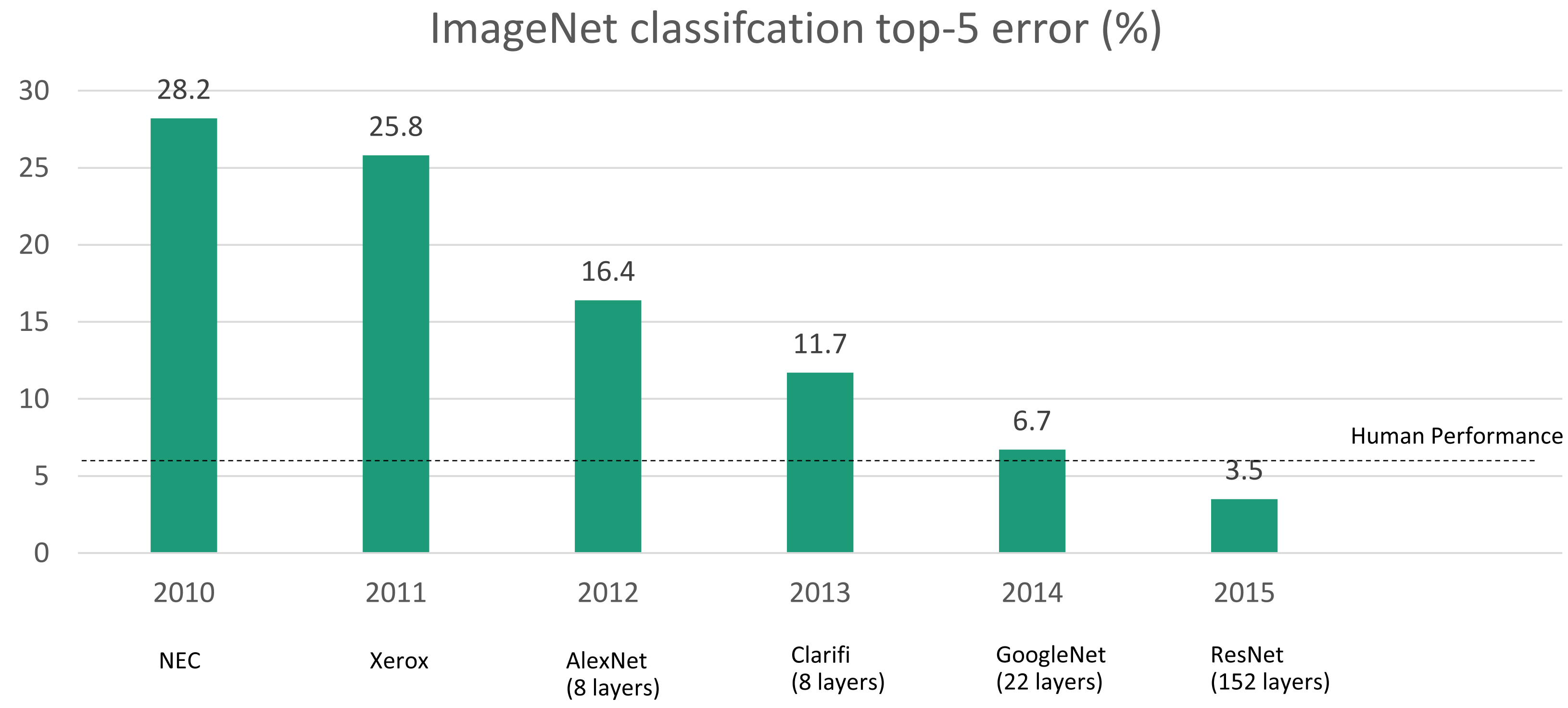
… and can be arranged in layers

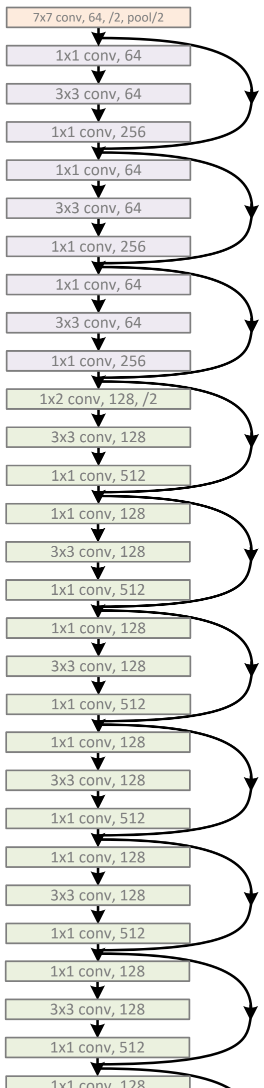


Revolution of depth

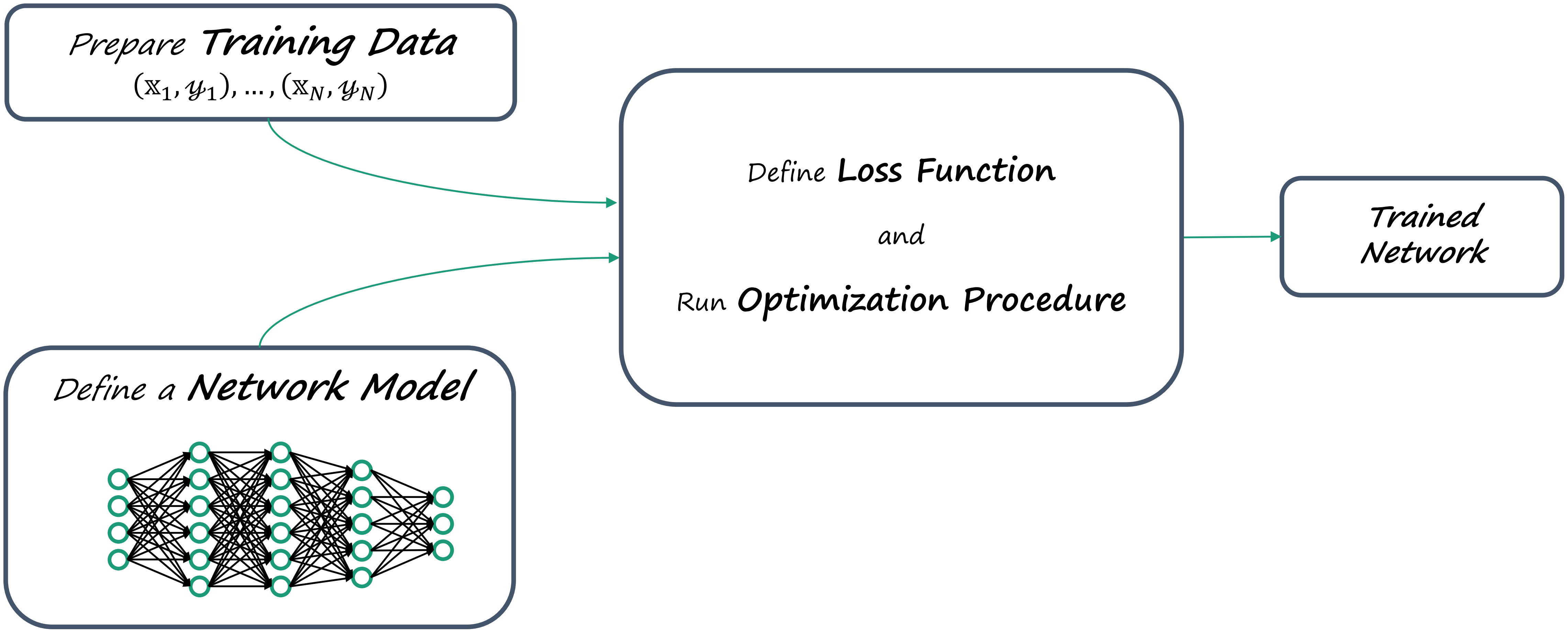
ResNet: 152 layers, and 1001 layers later on

MSRA's ResNet won the 1st places in ImageNet classification, ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation in [ILSVRC] & [COCO] competitions 2015

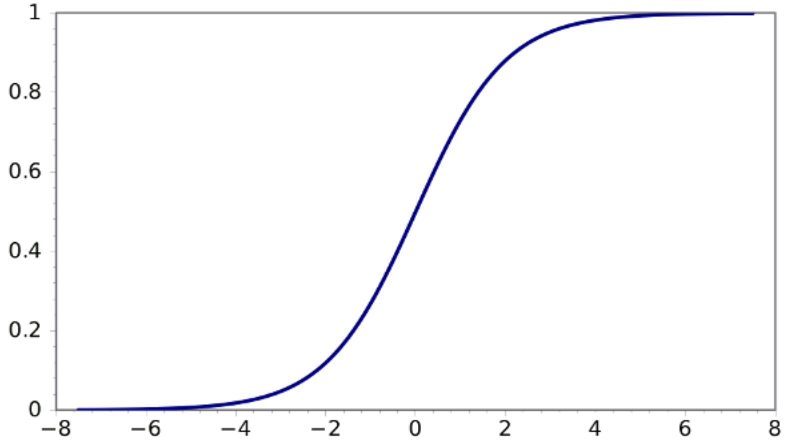


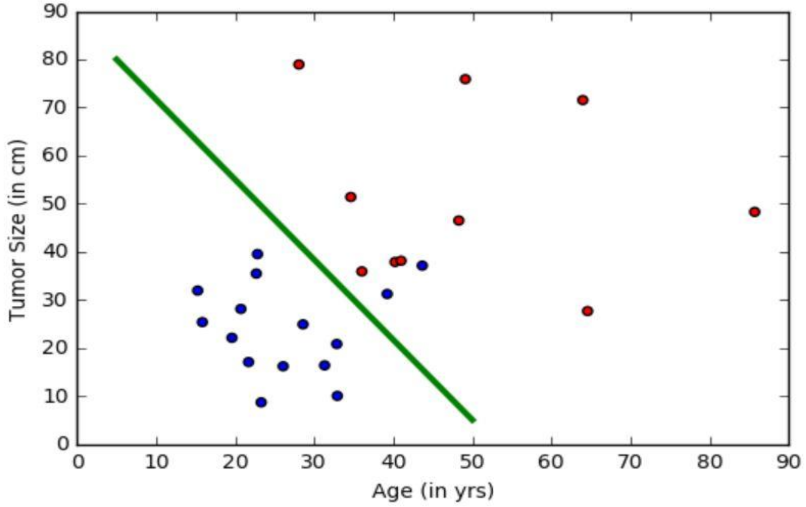


ANN Learning



Two class logistic regression – aka. single neuron ANN





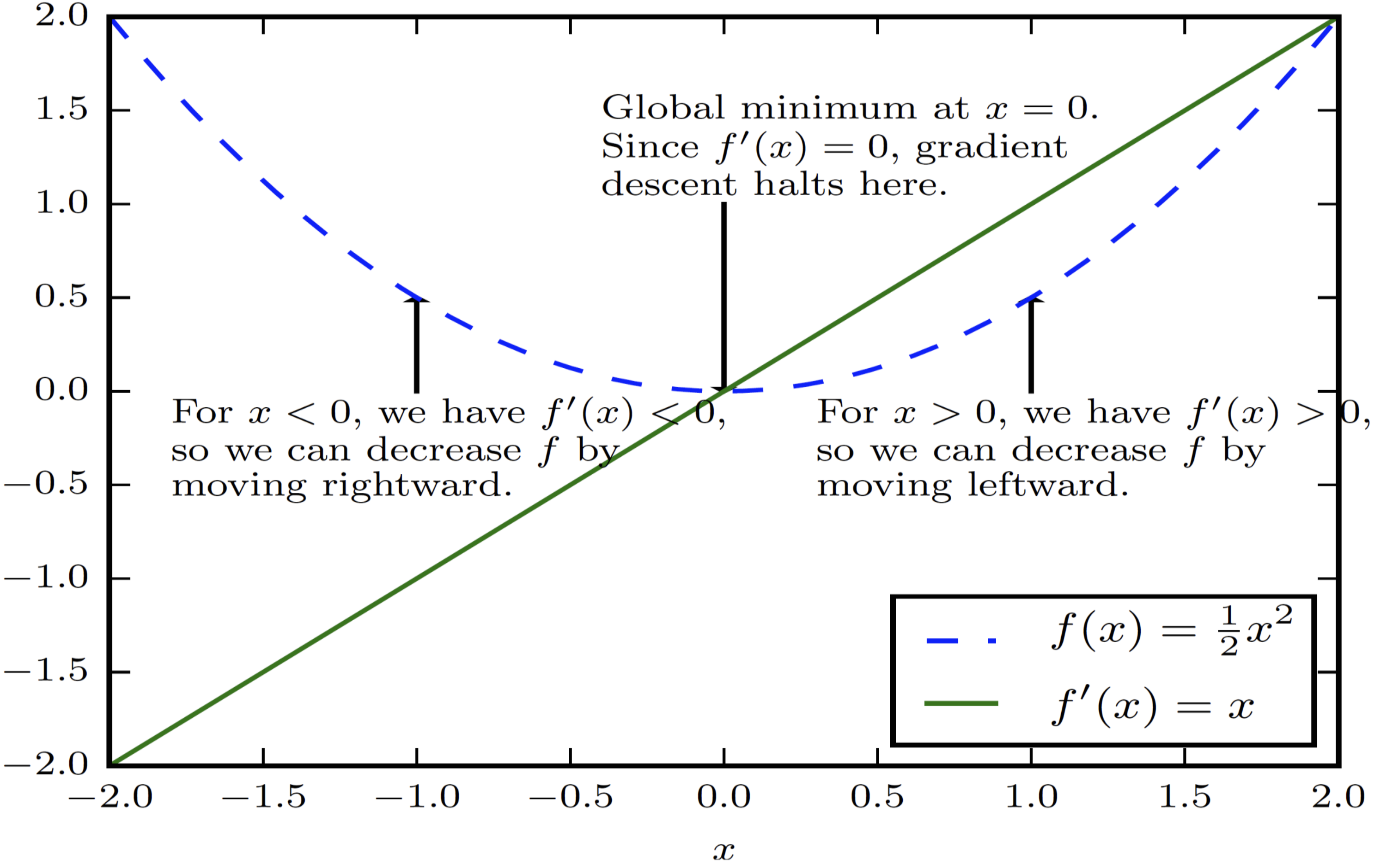
Training Data: $\boldsymbol{D} = \{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, \boldsymbol{y_2}), \ldots, (\boldsymbol{x_N}, \boldsymbol{y_N})\}$

Hypothesis Space: $\boldsymbol{h}(\boldsymbol{x}) = \dfrac{\mathbf{1}}{\mathbf{1} + \boldsymbol{e}^{-(\boldsymbol{w}^T \boldsymbol{x} + b)}}$

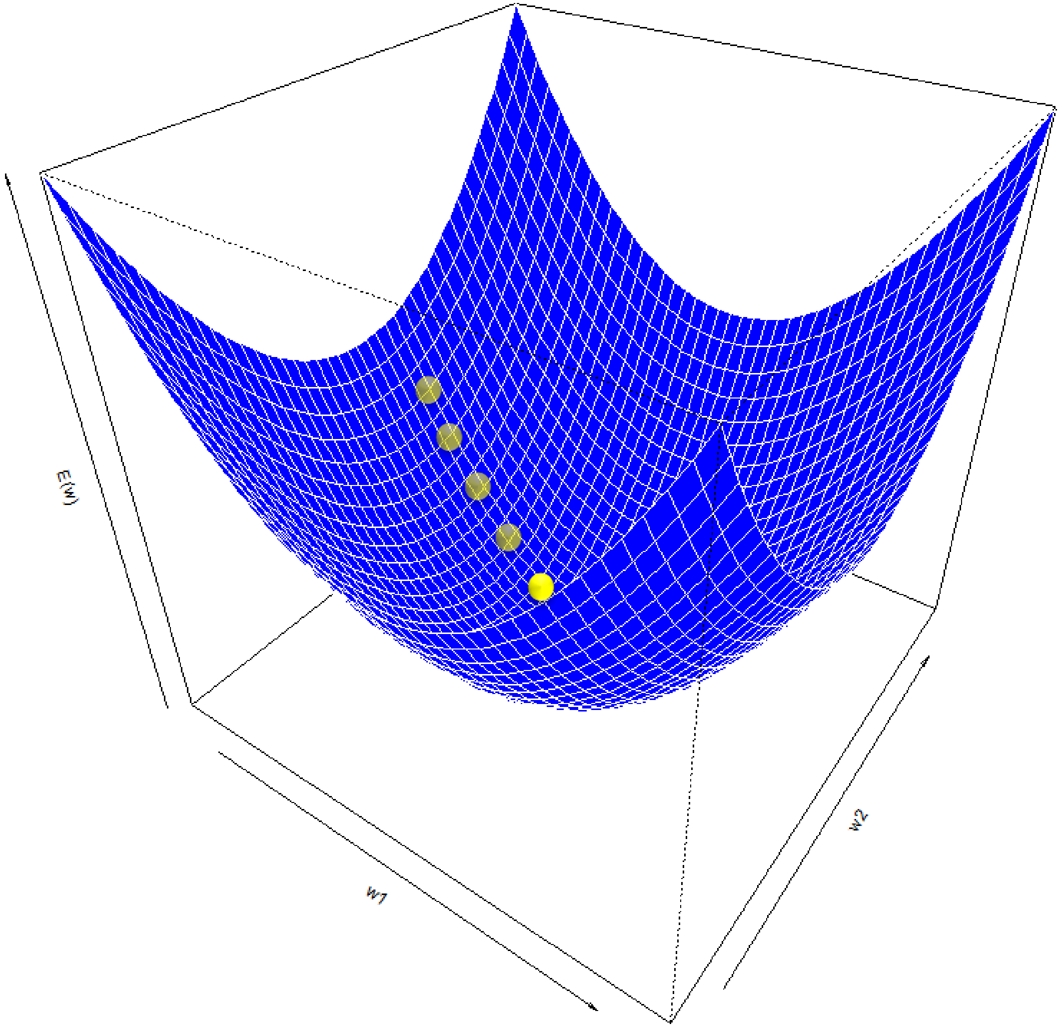
Loss Function: $E(w) = -\sum_{j=1}^{N} (y^{(j)} log\big(h(x^{(j)})\big) + \big(1 - y^{(j)}\big) log(1 - h(x^{(j)})))$

Optimization Procedure: ***Gradient Descent***

Gradient descent (Augustin Cauchy, 1847)



Gradient descent in two dimensional parameter space



$$E(w) = -\sum_{j=1}^{N}\left(y^{(j)} log\left(h(x^{(j)})\right) + \left(1 - y^{(j)}\right) log(1 - h(x^{(j)}))\right)$$

$$\nabla J(w) \equiv \left[\frac{\partial J}{\partial w_1}, \frac{\partial J}{\partial w_2}\right]$$

$$w = w - \eta \nabla J(w)$$

Stochastic Gradient Descent and Backpropagation

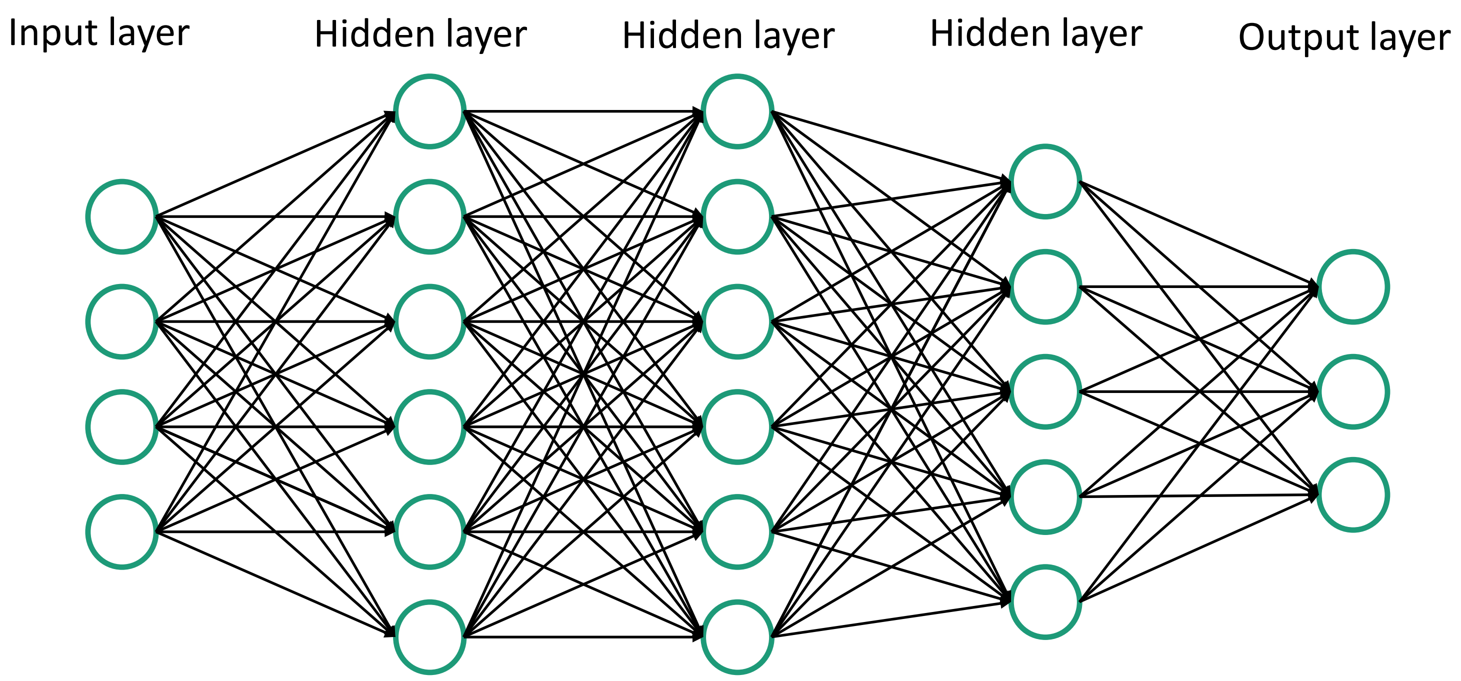
Loss function

$$\min_{\boldsymbol{w}} \sum_{i=1}^{N} f(x_i, y_i; w)$$

Stochastic Gradient Descent (SGD)

$$g(w_t) = \nabla f(x_i, y_i; w_t)$$

$$w_{t+1} = w_t - \eta_t g(w_t)$$



Stochastic Gradient Descent

Initialize learning rate $\boldsymbol{\eta}$
Initialize parameter vector $\boldsymbol{w}$
while stopping criterion not met **do**

Sample a minibatch of $\boldsymbol{m}$ examples from the training set

Compute gradient estimates: $\hat{\boldsymbol{g}} \leftarrow \frac{\mathbf{1}}{\boldsymbol{m}} \nabla_{\boldsymbol{w}} \sum_{\boldsymbol{i}=\mathbf{1}}^{\boldsymbol{m}} \boldsymbol{E}\left(\boldsymbol{w}, \boldsymbol{x}^{(\boldsymbol{i})}, \boldsymbol{y}^{(\boldsymbol{i})}\right)$

Apply update: $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \hat{\boldsymbol{g}}$

end

How do we calculate the gradient ?

The back propagation algorithm

1. **Input** x:
   Set the activations for the input layer $l = 1$

2. **Feedforward:**
   For each layer $l = 2,3,\ldots,L$ compute: $z^l = w^l a^{l-1} + b^l \; and \; a^l = \sigma(z^l)$

3. **Output error** δ^L:
   Compute the vector: $\delta^L = \nabla_a E \odot \sigma'(z^L)$

4. **Backpropagate the error:**
   For each layer $l = L-1, L-2, \ldots, 2$ $\delta^l = \left(\left(w^{l+1}\right)^T \delta^{l+1}\right) \odot \sigma'(z^l)$

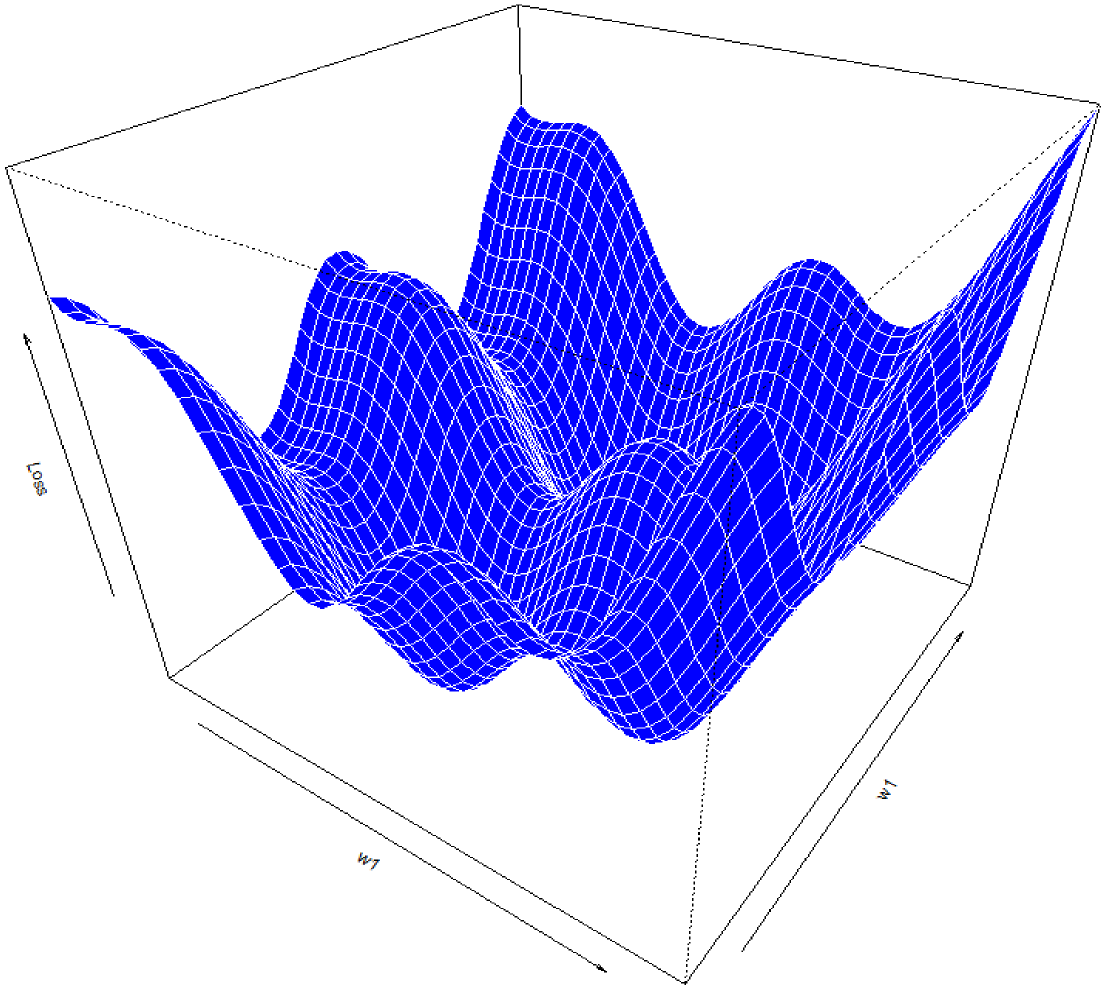
5. **Calculate gradient:**
   $$\frac{\partial E}{w^l_{jk}} = a^{l-1}_k \delta^l_j \; and \; \frac{\partial E}{b^l_j} = \delta^l_j$$

Loss functions and Deep Learning

- ML literature and toolkits use different names for the same concept:
  - Objective function, loss function, cost function, error function, **criterion**
- In Deep Learning we often optimize a criterion which is not the same as the performance measure *P* we care about
  - Since optimizing the direct performance measure may be intractable,
  - One optimizes a **surrogate loss function**
- An example, common loss function for multinomial classification is **cross-entropy**

$$E(w) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_j \, log\left(p_j\right)$$

Deep Learning reality - non-convex loss functions



Imagine that in 1,000,000 dimensional space

Deep Learning optimization procedures

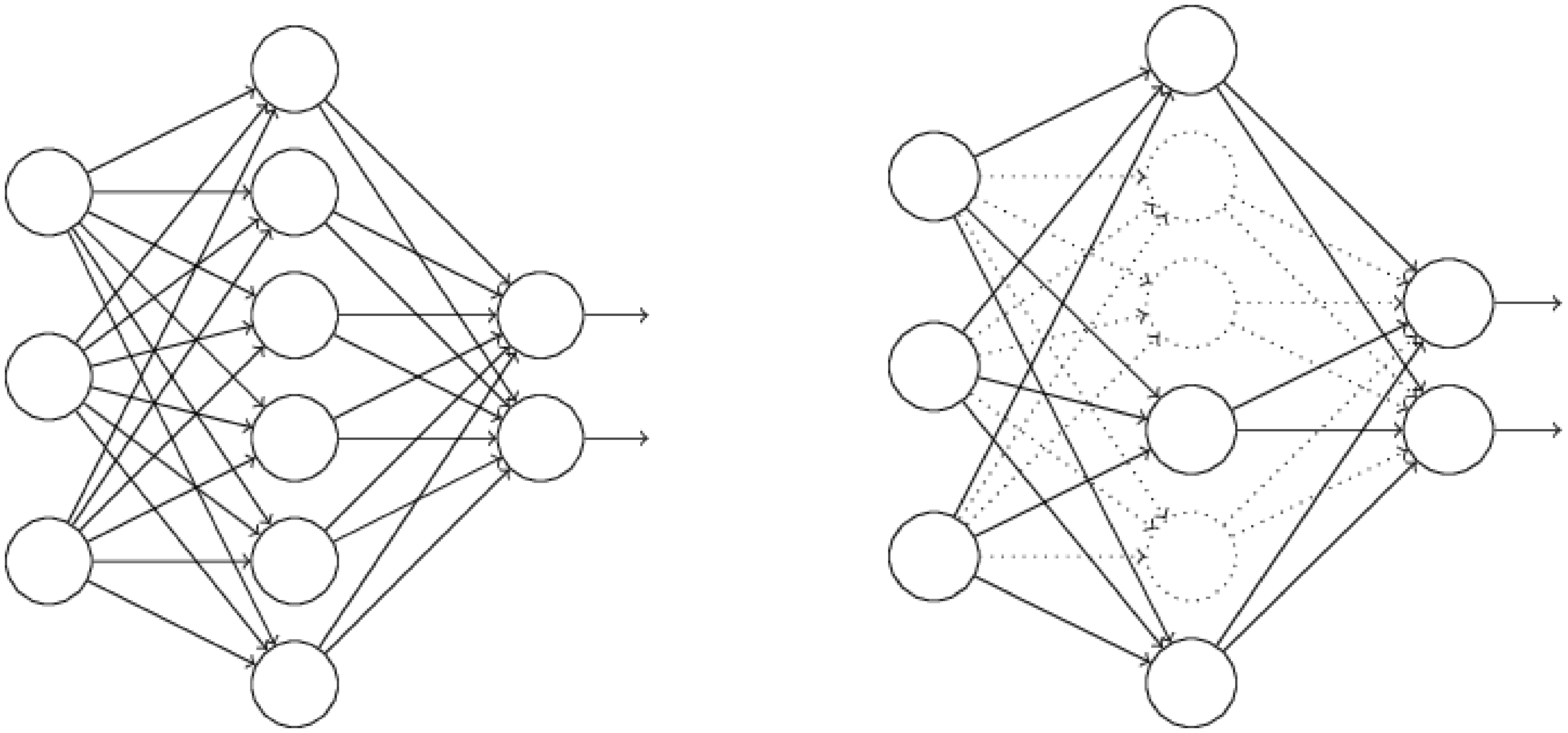
- Stochastic Gradient Descent – SDG
- SDG with momentum
- AdaGrid
- RMSProp
- Adam
- Newton’s Method
- Conjugate Gradients
- BFGS

Preventing overfitting in Neural Networks

- Regularization
  - L2 regularization $E(w) = E(w)_0 + \frac{\lambda}{2n}\sum_w w^2$
  - L1 regularization $E(w) = E(w)_0 + \frac{\lambda}{n}\sum_w |w|$
- Early stopping
- Data augmentation

- Dropout

Dropout



Descending the rugged terrain of a multidimensional world can be a scary adventure

… You will be disoriented by countless saddle points

… You may get lost in vast, sparse, plateaus

… You risk falling down deep cliffs

… You could be injured by exploding gradients

Sources and acknowledgements

- Frank Seide (2017). Training Deep Models Like Microsoft Product Groups.
- Christopher Bishop (2006). Pattern Recognition and Machine Learning.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville (2016). Deep Learning.
- Michael Nielsen (2017). Neural Networks and Deep Learning.
- Yaser Abu-Mostafa et al. (2012) Learning From Data.
- Trevor Hastie et al. (2008). The Elements of Statistical Learning.
- Tom M. Mitchell (1997). Machine Learning.