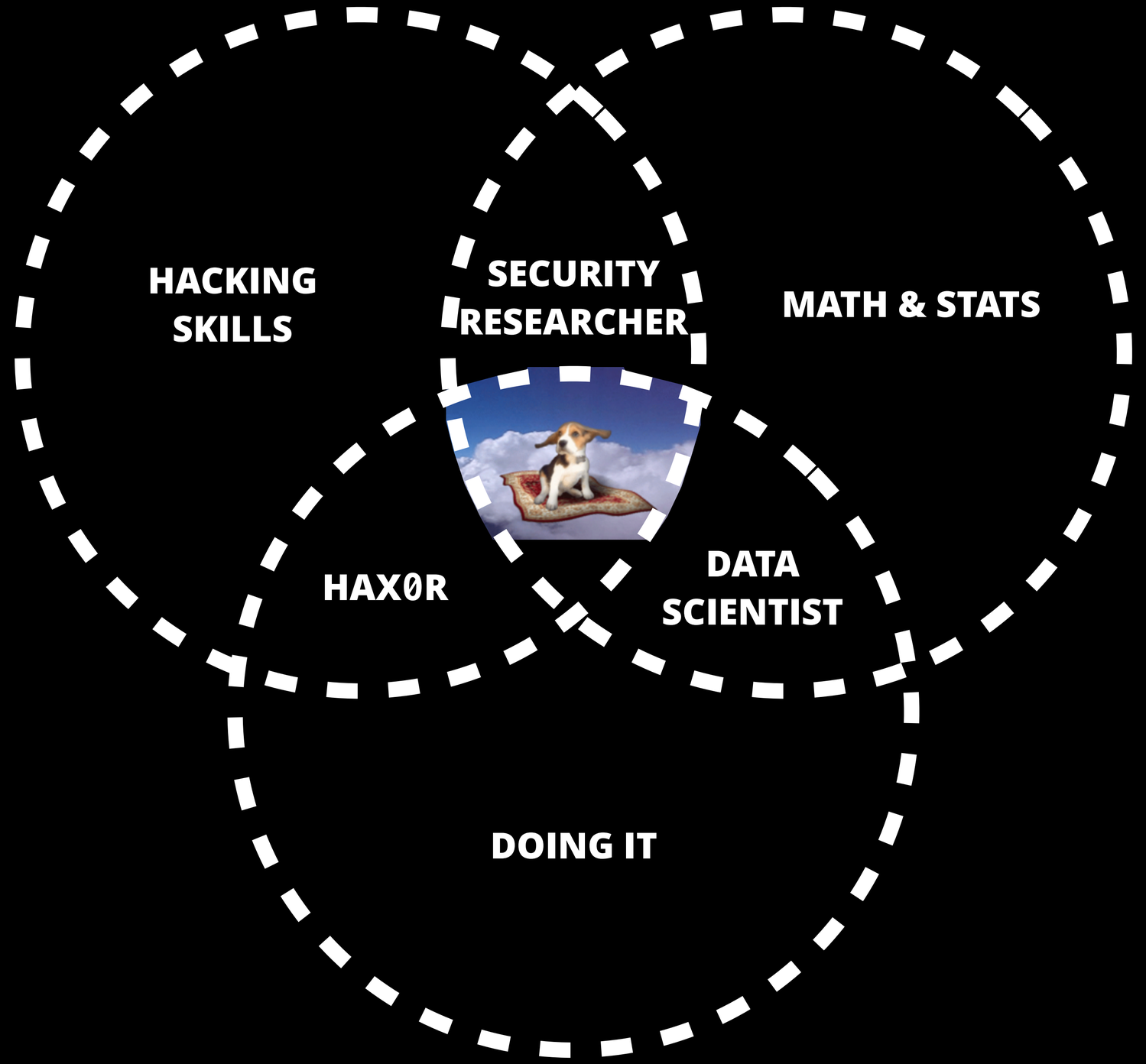
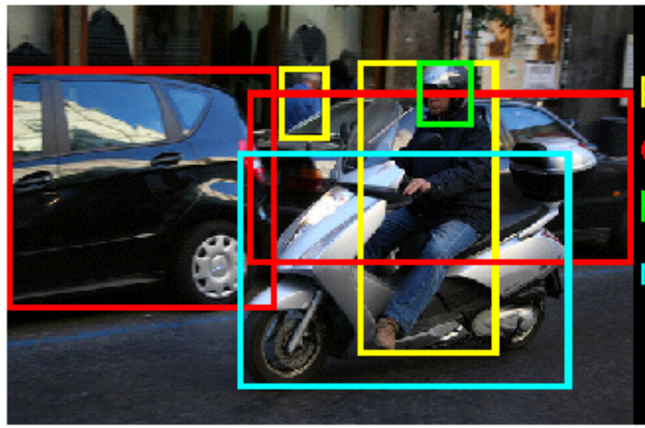
Machine Duping
Pwning Deep Learning Systems



CLARENCE CHIO
MLHACKER
@cchio



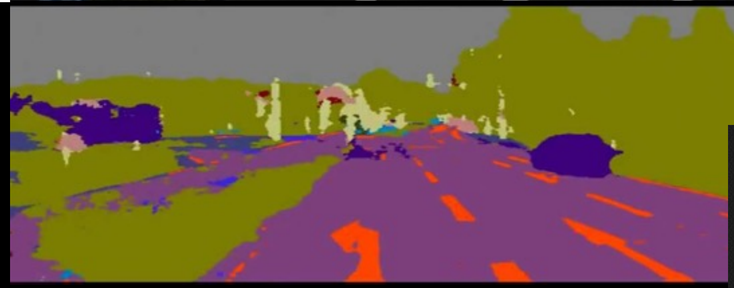
adapted from
Dave Conway



ImageNet

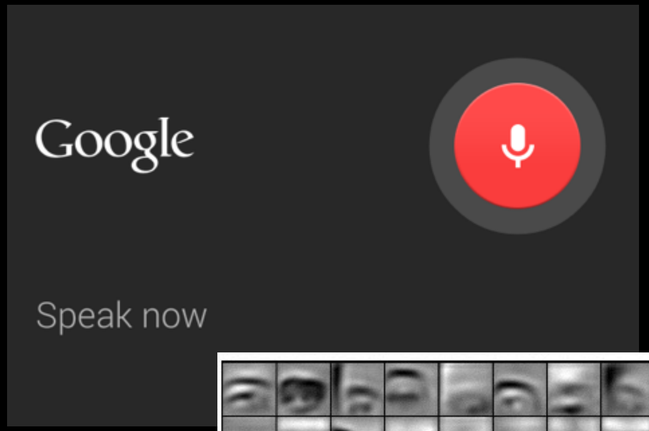
Nvidia DRIVE



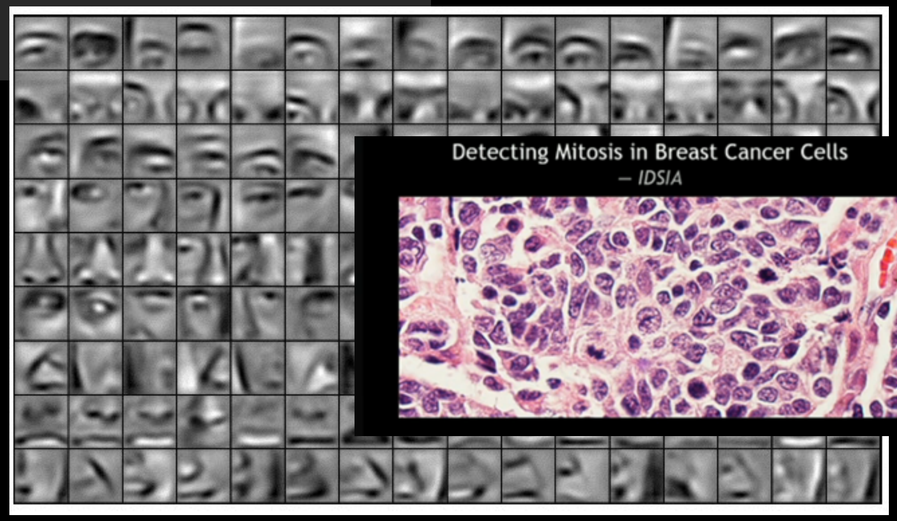


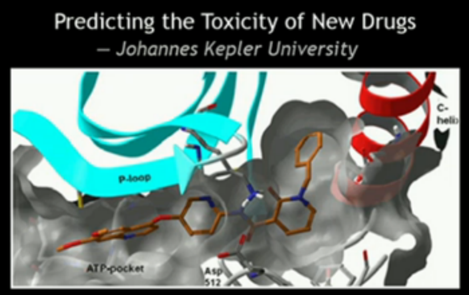


Google DeepMind



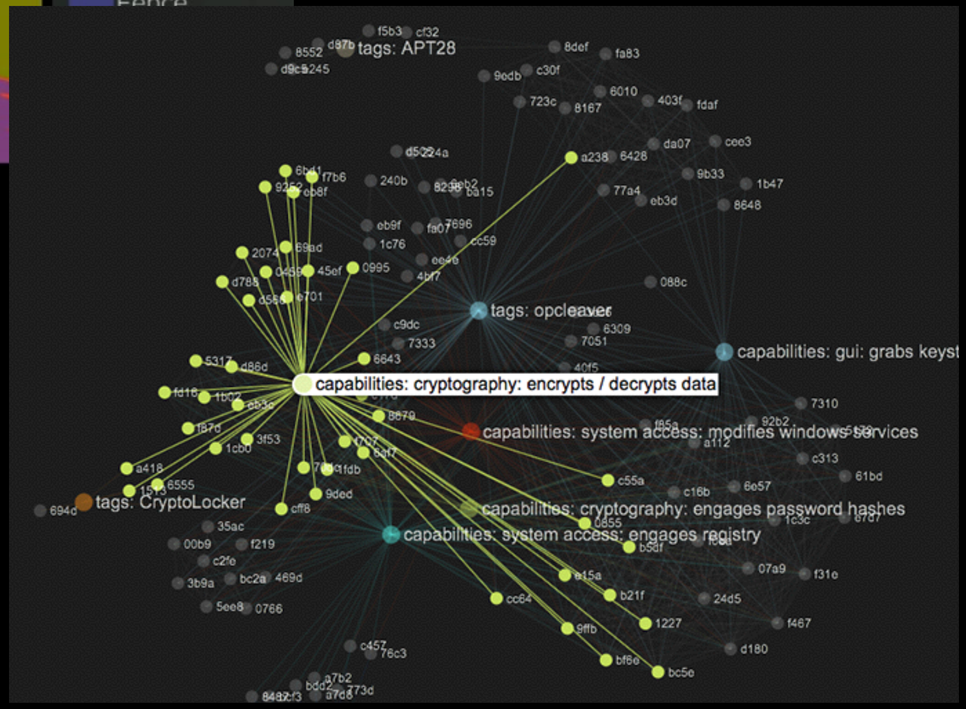
Google Inc.





Nvidia

UMass, Facial Recognition



Invincea Malware Detection System

Deep Learning

why would someone choose to use it?

(Semi)Automated
feature/representation
learning

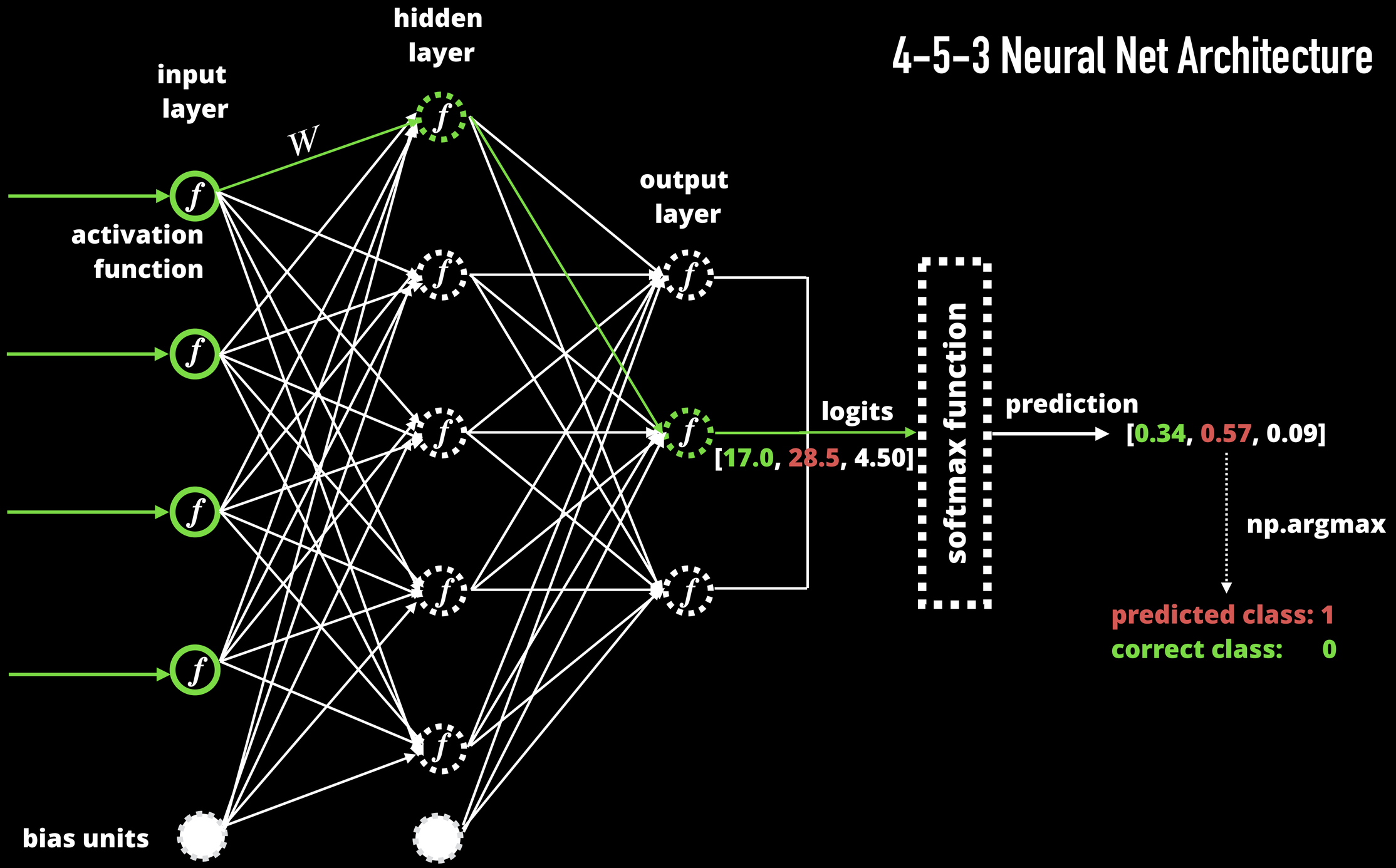
One infrastructure for
multiple problems (sort of)

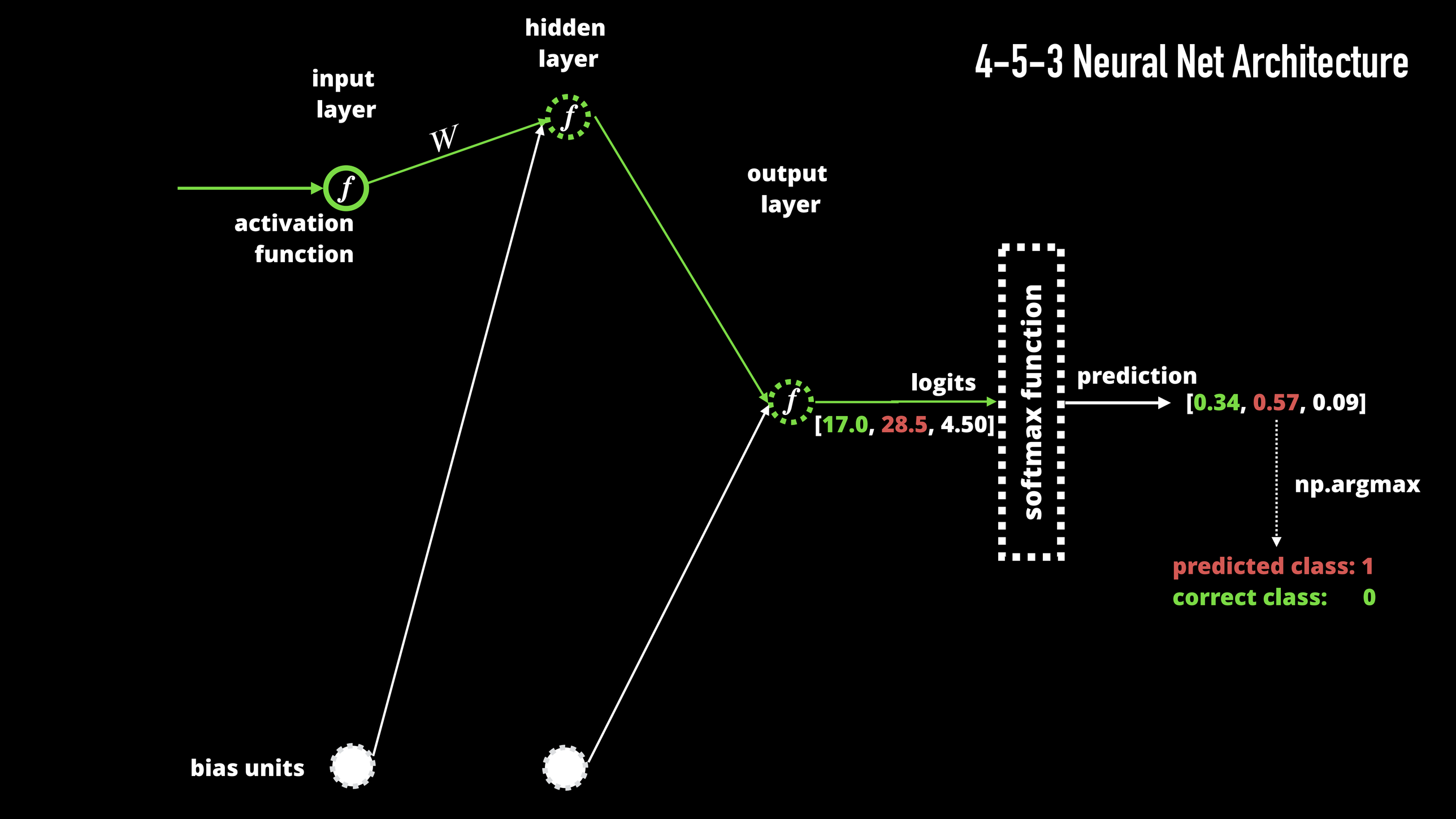
Hierarchical learning:
Divide task across
multiple layers

Efficient,
easily distributed &
parallelized

Definitely not one-size-fits-all

4–5–3 Neural Net Architecture

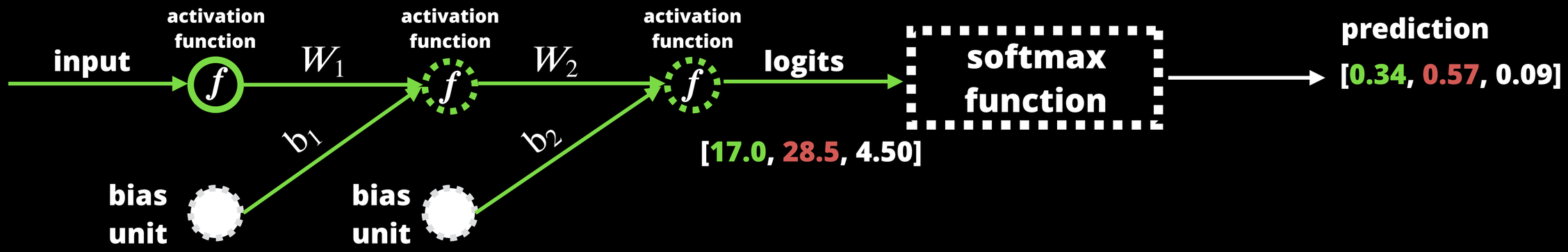



4–5–3 Neural Net Architecture
input layer
hidden layer
output layer
W
f
activation function
bias units
logits
[17.0, 28.5, 4.50]
softmax function
prediction
[0.34, 0.57, 0.09]
np.argmax
predicted class: 1
correct class: 0

Training Deep Neural Networks

Step 1 of 2: Feed Forward

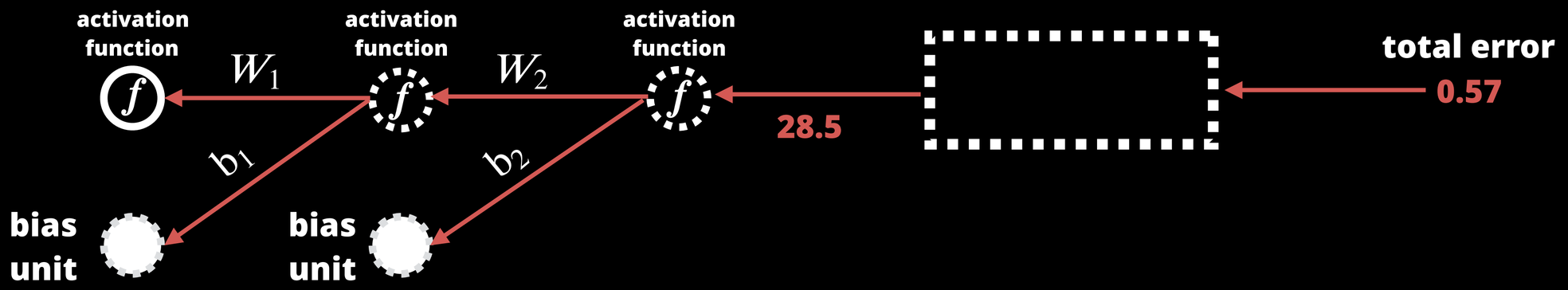
1. Each unit receives output of the neurons in the previous layer (+ bias signal)
2. Computes weighted average of inputs
3. Apply weighted average through nonlinear activation function
4. For DNN classifier, send final layer output through softmax function



Training Deep Neural Networks

Step 2 of 2: Backpropagation

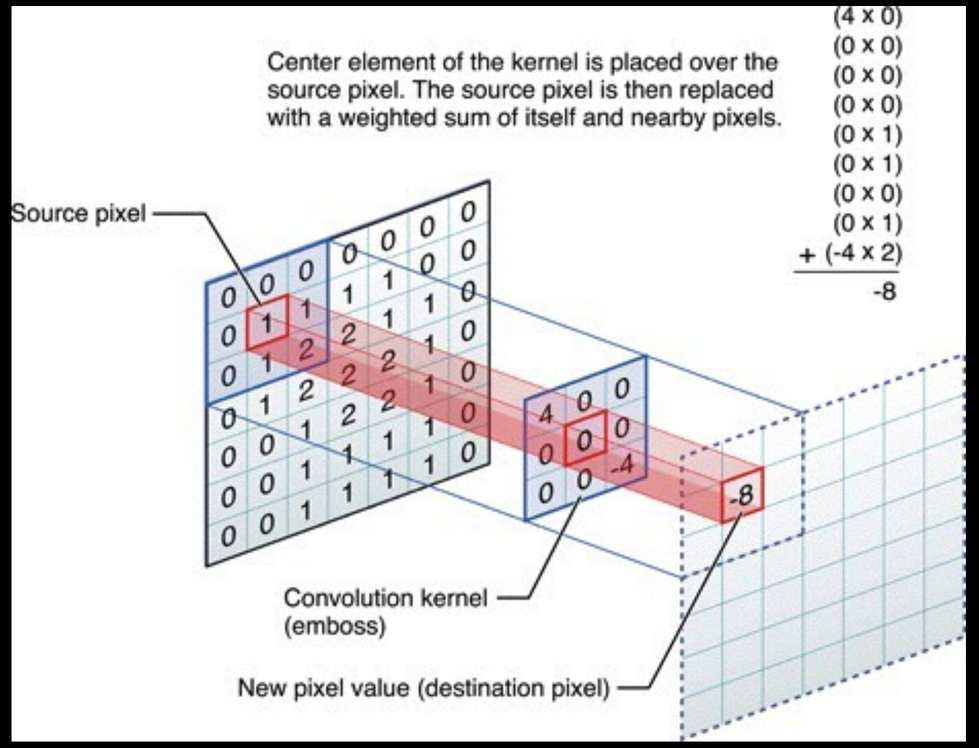
1. **If the model made a wrong prediction, calculate the error**
    1. In this case, the correct class is 0, but the model predicted 1 with 57% confidence - error is thus 0.57
2. **Assign blame: trace backwards to find the units that contributed to this wrong prediction (and how much they contributed to the total error)**
    1. Partial differentiation of this error w.r.t. the unit's activation value
3. **Penalize those units by decreasing their weights and biases by an amount proportional to their error contribution**
4. **Do the above efficiently with optimization algorithm e.g. Stochastic Gradient Descent**



DEMO

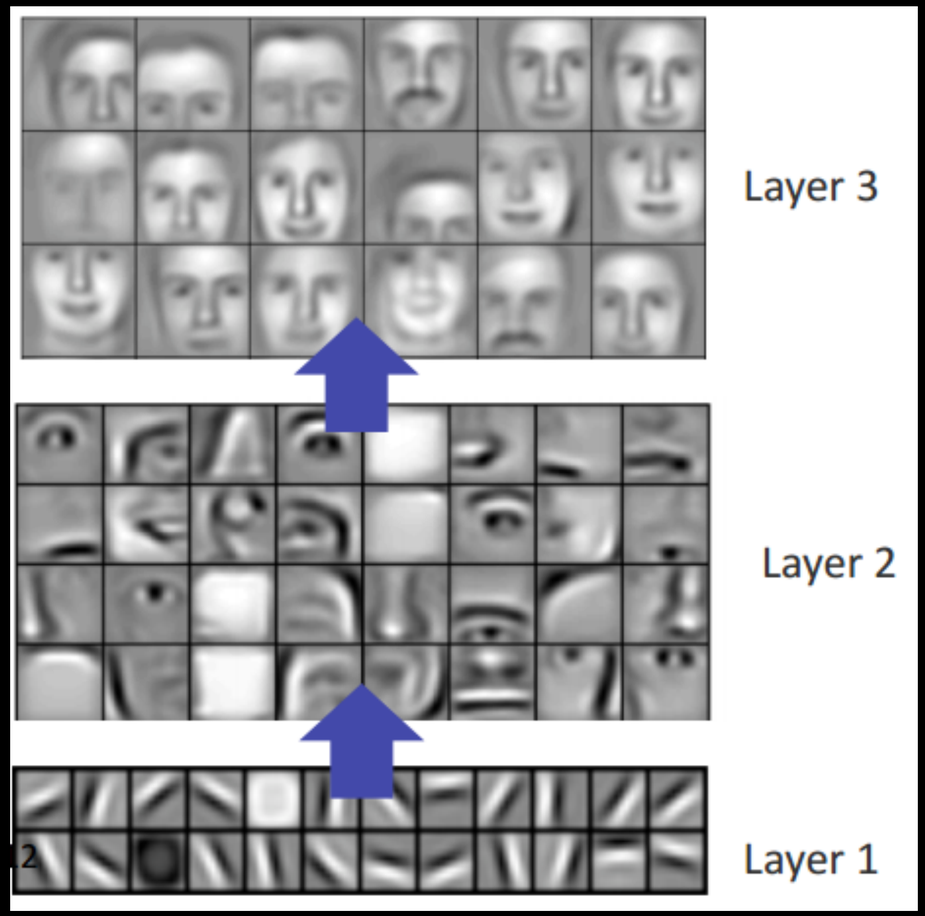
Beyond Multi Layer Perceptrons

Convolutional Neural Network



Source: iOS Developer Library
vImage Programming Guide

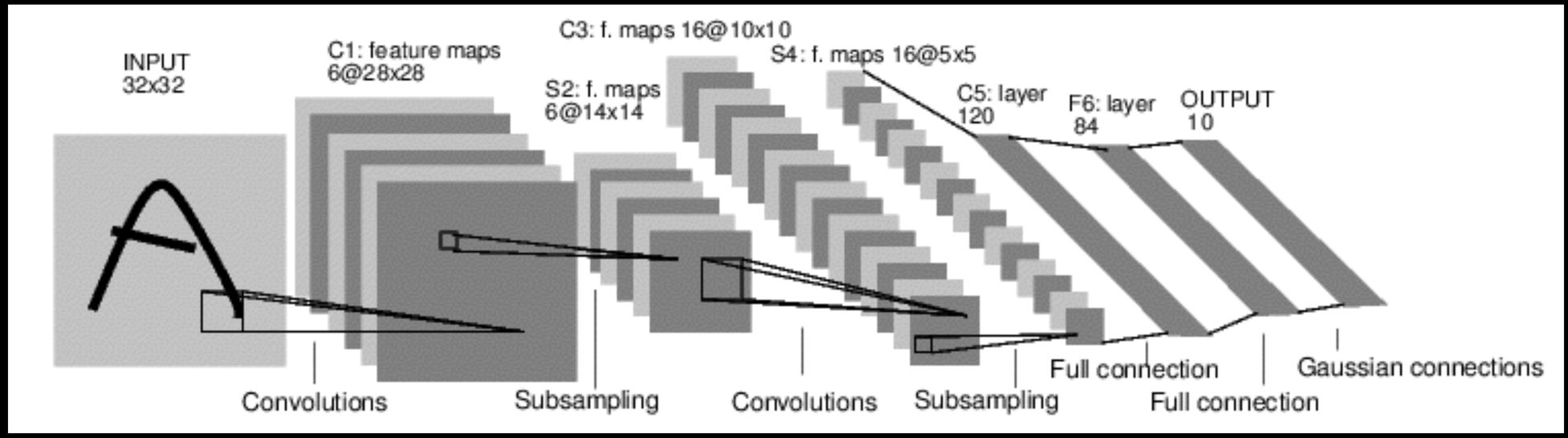
Layered Learning



Lee, 2009, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations."

Beyond Multi Layer Perceptrons

Convolutional Neural Network

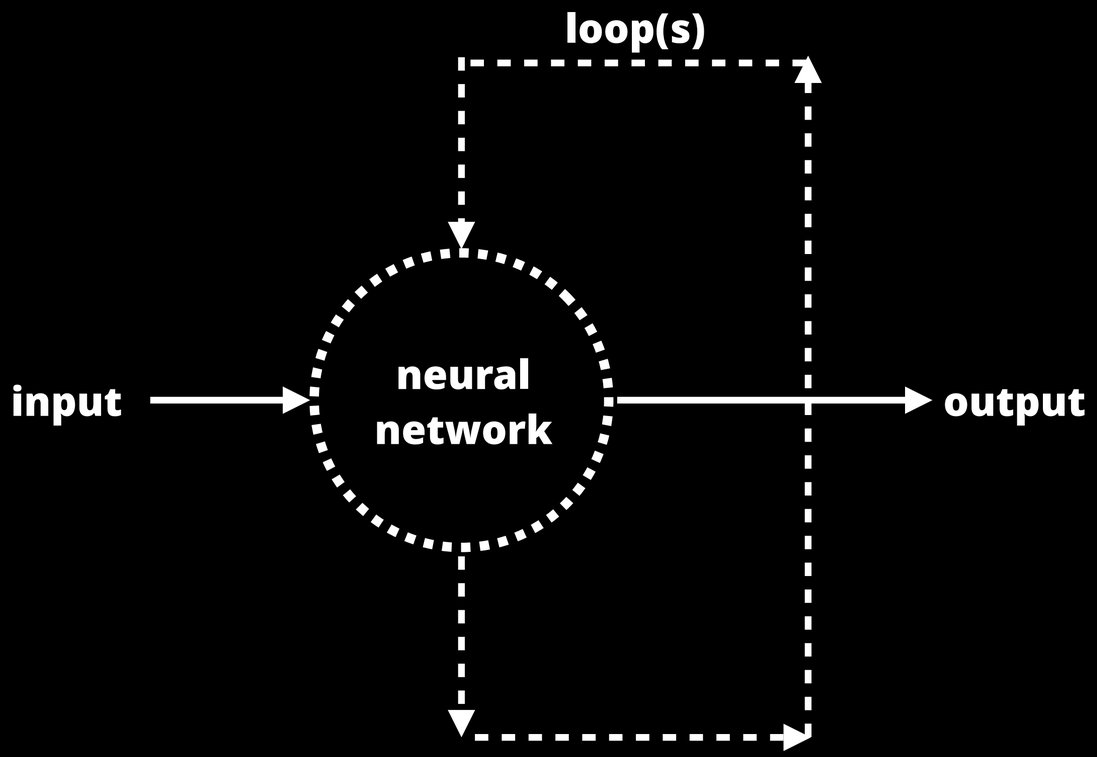


Source: LeNet 5, LeCun et al.

Beyond Multi Layer Perceptrons

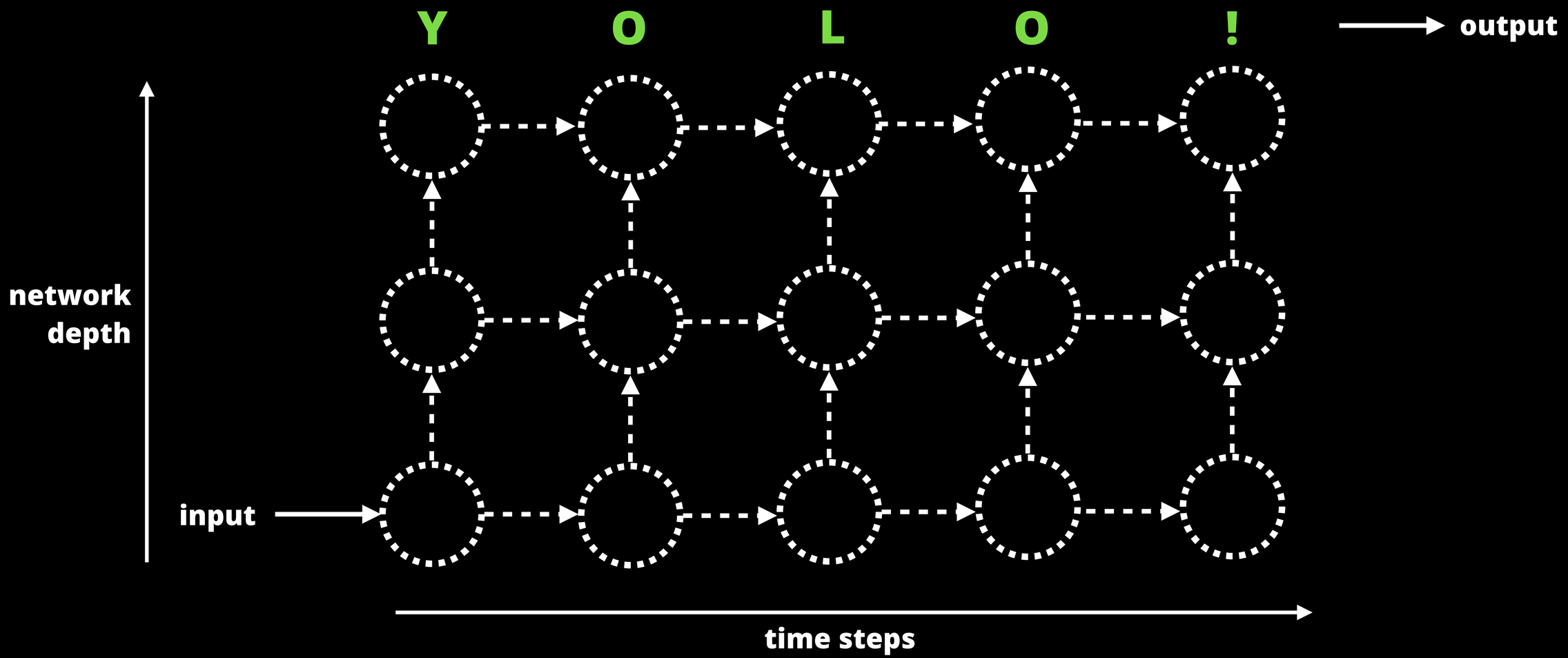
Recurrent Neural Network

- Just a DNN with a feedback loop
- Previous time step feeds all intermediate and final values into next time step
- Introduces the concept of “memory” to neural networks



Beyond Multi Layer Perceptrons

Recurrent Neural Network

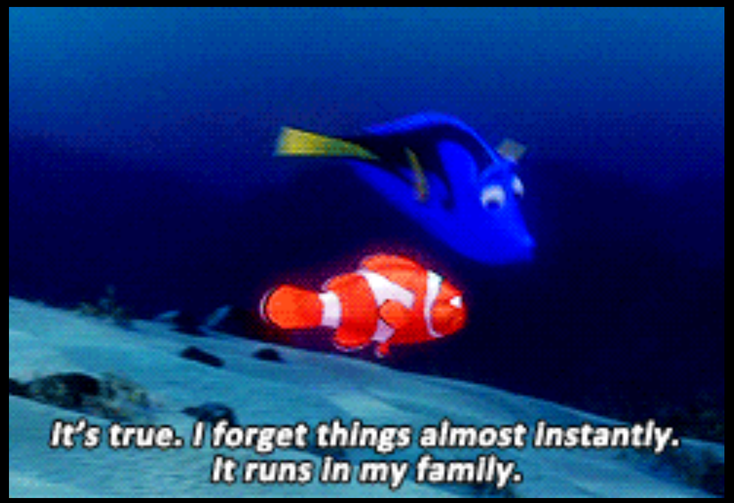


Beyond Multi Layer Perceptrons

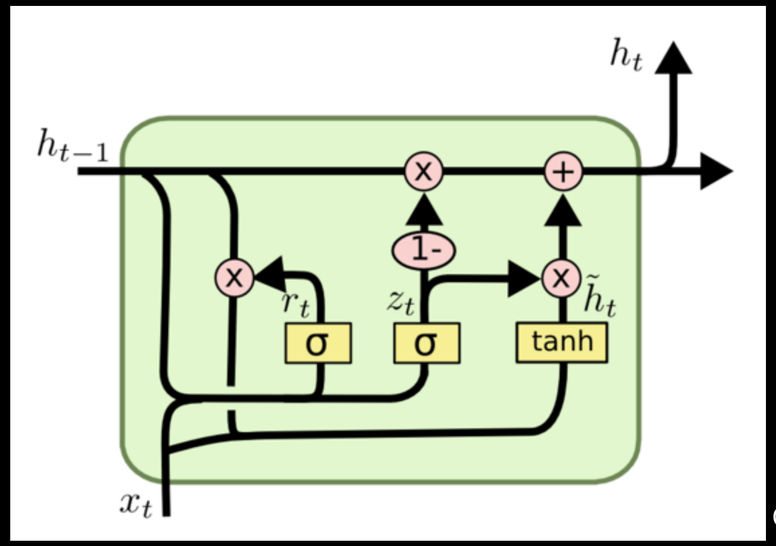
Long Short-Term Memory (LSTM) RNN

- To make good predictions, we sometimes need more context

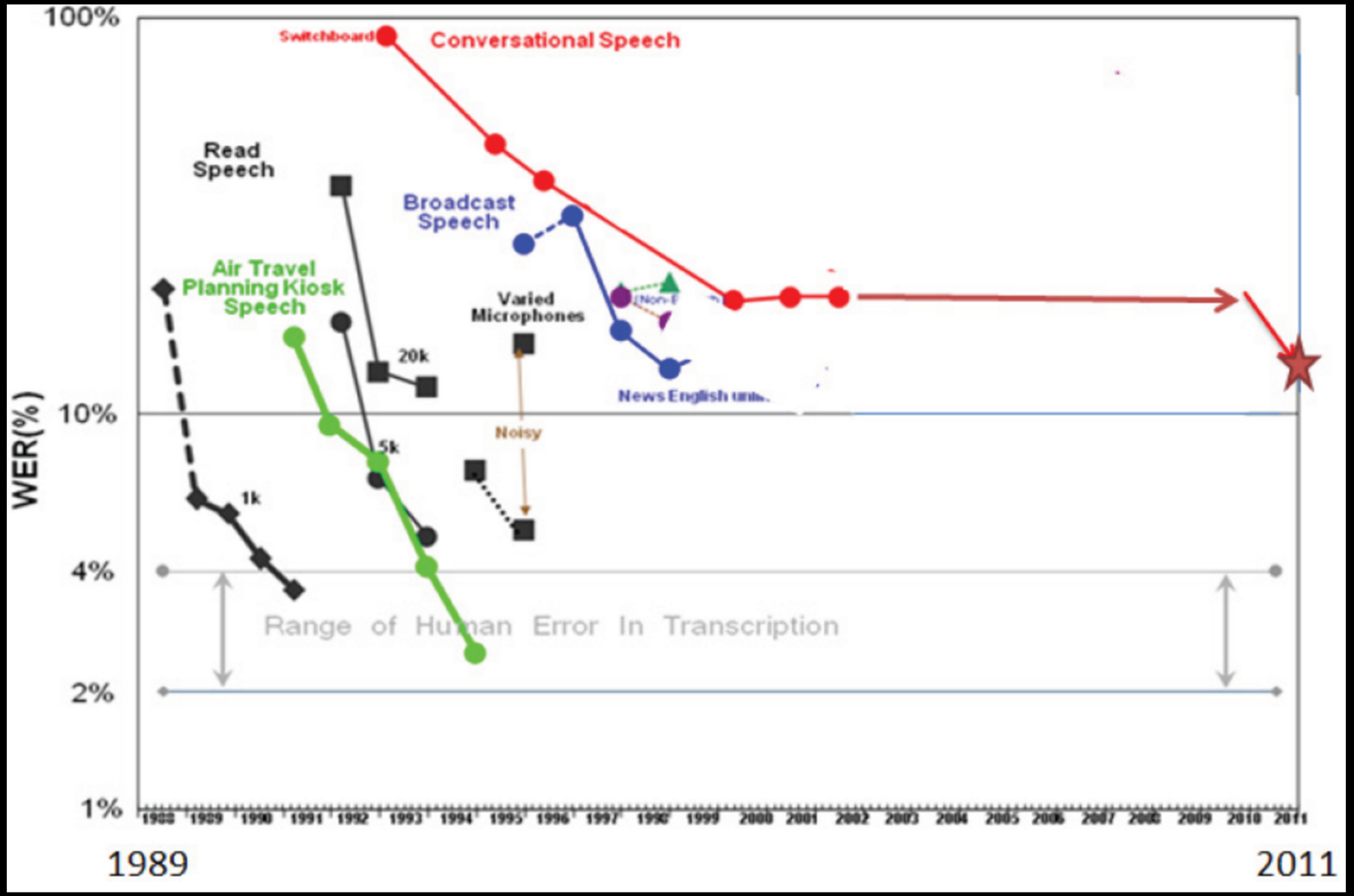
- We need **long-term** memory capabilities without extending the network's recursion indefinitely (unscalable)



Disney, Finding Dory



Colah's Blog, "Understanding LSTM Networks"



Deng et al. “Deep Learning: Methods and Applications”

HOW TO PWN?

Attack Taxonomy

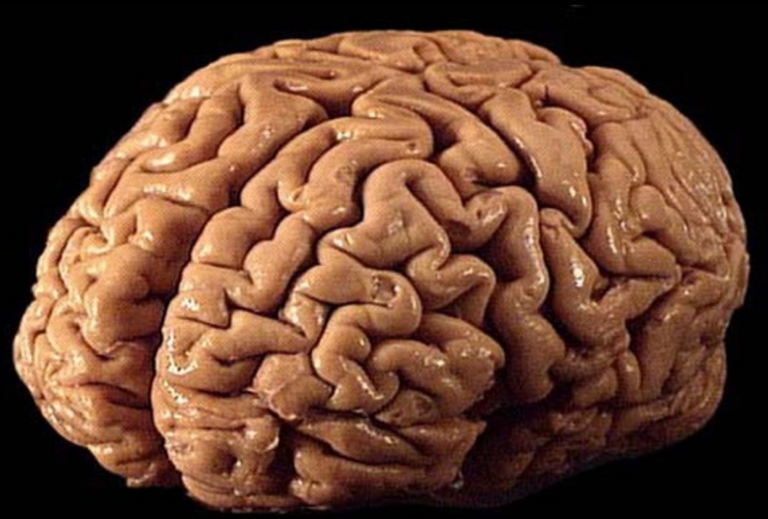
Applicable also to **online-learning** models that continuously learn from real-time test data

	Causative (Manipulative training samples)	**Exploratory** (Manipulative test samples)
Targeted	Training samples that move classifier decision boundary in an intentional direction	Adversarial input crafted to cause an intentional misclassification
Indiscriminate	Training samples that increase FP/FN → renders classifier unusable	n/a

Why can we do this?



vs.



BLINDSPOTS:

Statistical learning models don’t **learn** concepts the same way that we do.

Adversarial Deep Learning

Intuitions

1. **Run input *x* through the classifier model (or substitute/approximate model)**
2. **Based on model prediction, derive a perturbation tensor that maximizes chances of misclassification:**
    1. Traverse the manifold to find blind spots in input space; or
    2. Linear perturbation in direction of neural network's cost function gradient; or
    3. Select only input dimensions with high saliency* to perturb by the model's Jacobian matrix
3. **Scale the perturbation tensor by some magnitude, resulting in the effective perturbation (δ_x) to *x***
    1. Larger perturbation == higher probability for misclassification
    2. Smaller perturbation == less likely for human detection

*** saliency:** amount of **influence** a selected dimension has on the entire model's output

Adversarial Deep Learning

Intuitions

Szegedy, 2013: Traverse the manifold to find blind spots in the input space

- Adversarial samples == pockets in the manifold
- Difficult to efficiently find by brute force (high dimensional input)
- Optimize this search, take gradient of input w.r.t. target output class

Adversarial Deep Learning

Intuitions

Goodfellow, 2015: Linear adversarial perturbation

- Developed a linear view of adversarial examples
- Can just take the cost function gradient w.r.t. the sample (*x*) and original predicted class (*y*)
- Easily found by backpropagation

Adversarial Deep Learning

Intuitions

Papernot, 2015: Saliency map + Jacobian matrix perturbation

- More complex derivations for why the Jacobian of the learned neural network function is used
    - Obtained with respect to input features rather than network parameters
    - Forward propagation is used instead of backpropagation
- To reduce probability of human detection, only perturb the dimensions that have the greatest impact on the output (salient dimensions)

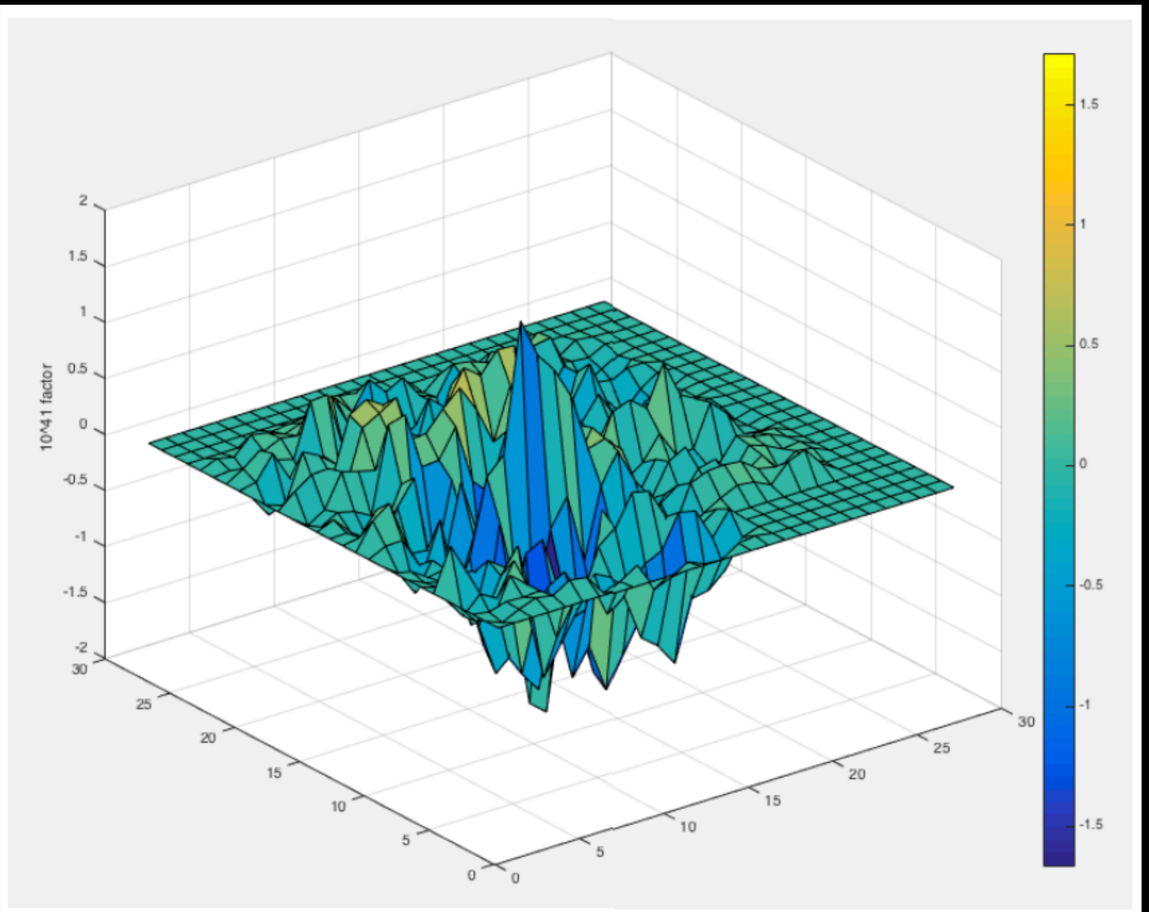
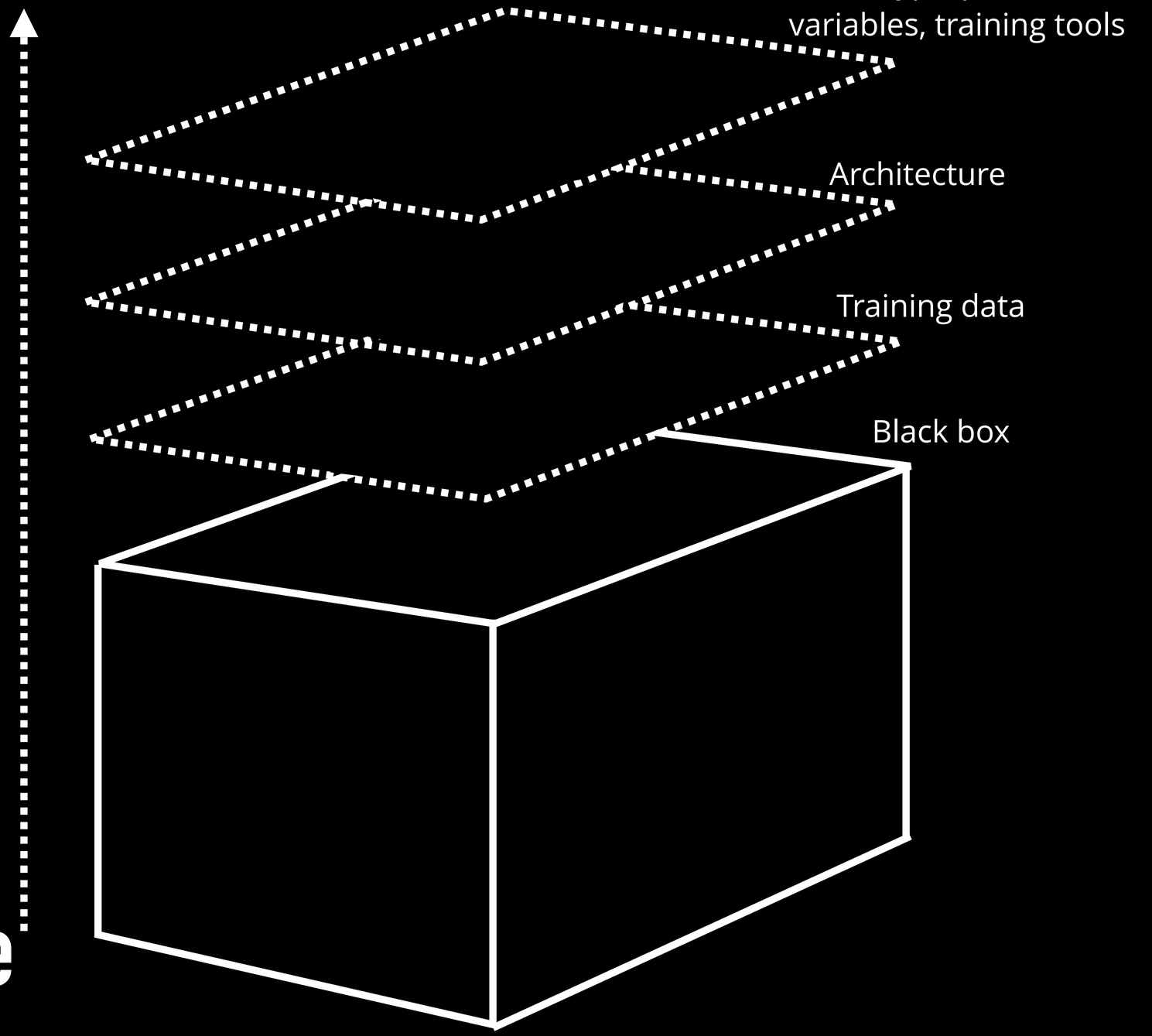


Fig. 7: Saliency map of a 784-dimensional input to the LeNet architecture (cf. validation section). The 784 input dimensions are arranged to correspond to the 28x28 image pixel alignment. Large absolute values correspond to features with a significant impact on the output when perturbed.

Papernot et al., 2015

Threat Model: Adversarial Knowledge

Increasing attacker knowledge



Deep Neural Network Attacks



What can you do with limited knowledge?

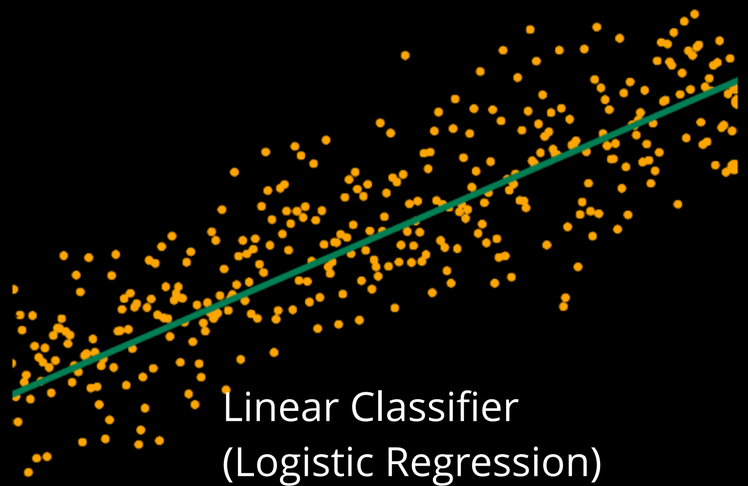
- Quite a lot.
- Make good guesses: Infer the methodology from the task
  - Image classification: ConvNet
  - Speech recognition: LSTM-RNN
  - Amazon ML, ML-as-a-service etc.: Shallow feed-forward network
- What if you can’t guess?

STILL CAN PWN?

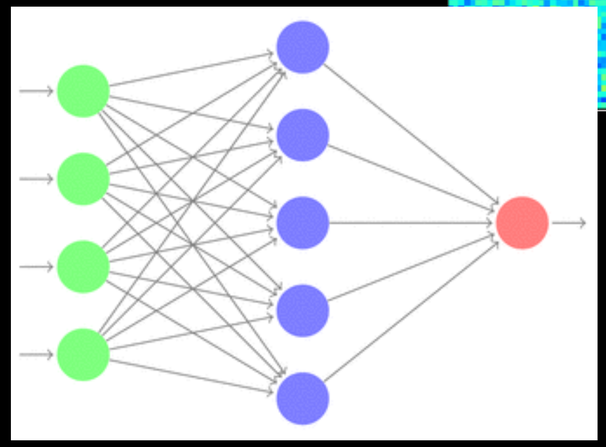
Black box attack methodology

1. Transferability

Adversarial samples that fool model A have a good chance of fooling a previously unseen model B

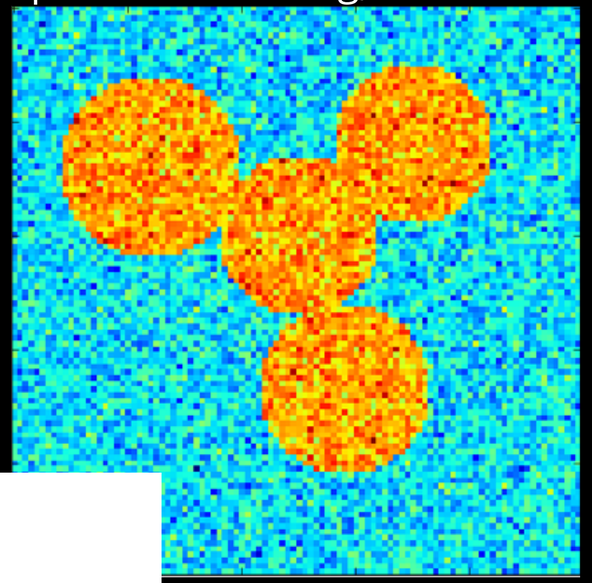


Linear Classifier
(Logistic Regression)

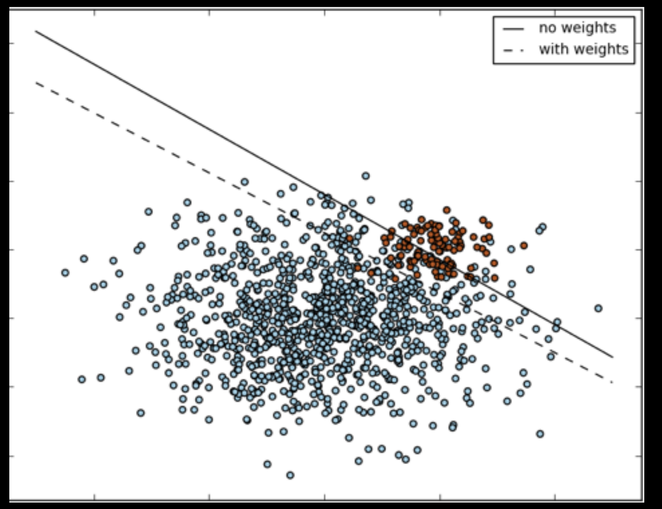


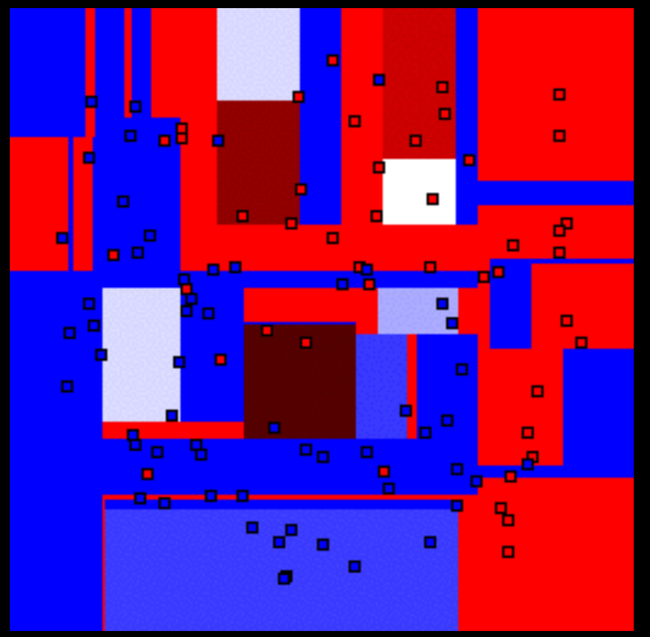
Feed Forward Neural Network

Spectral Clustering, scikit-learn



SVM, scikit-learn





Decision Tree
Matt's Webcorner, Stanford

Black box attack methodology

1. Transferability

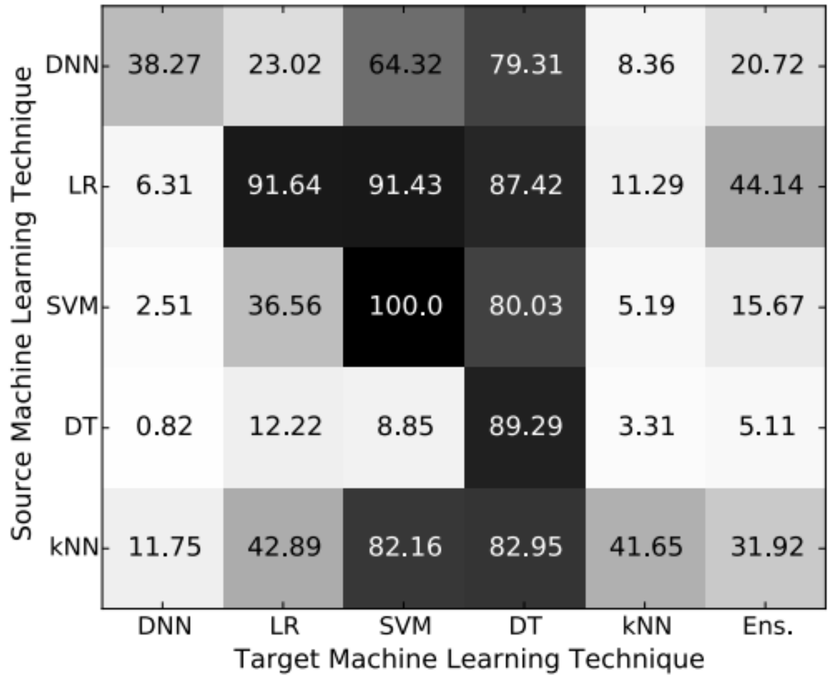


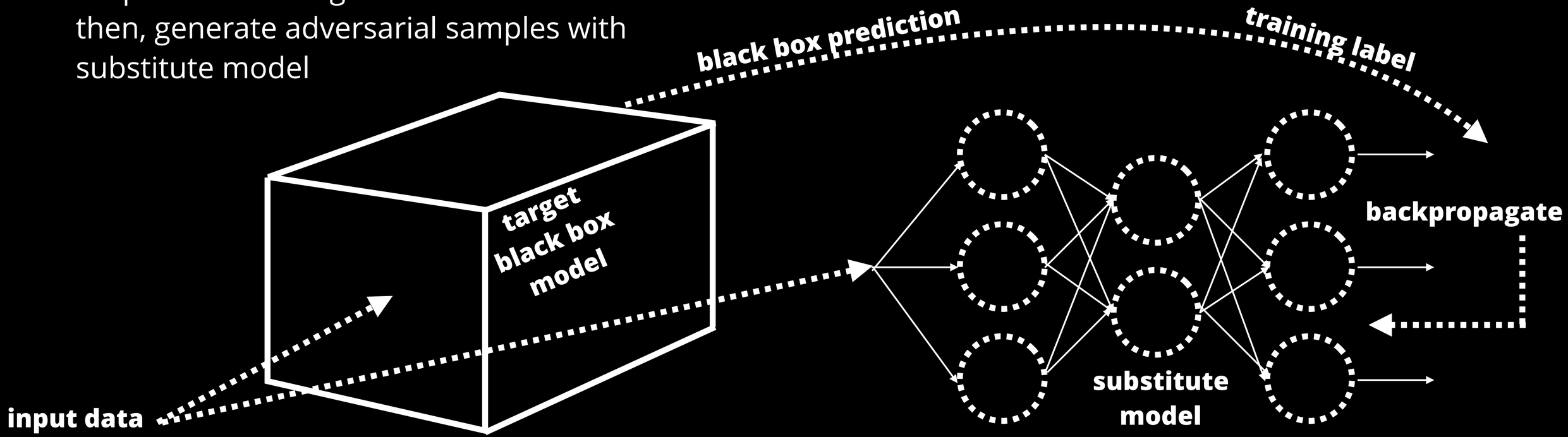
Figure 3: cross-technique Transferability matrix: cell (i, j) is the percentage of adversarial samples crafted to mislead a classifier learned using machine learning technique i that are misclassified by a classifier trained with technique j.

Papernot et al. “Transferability in Machine Learning:
from Phenomena to Black-Box Attacks using Adversarial Samples”

Black box attack methodology

2. Substitute model

train a new model by treating the target model's output as a training labels
then, generate adversarial samples with substitute model



Why is this possible?

- Transferability?
  - Still an open research problem
- Manifold learning problem
  - Blind spots
  - Model vs. Reality dimensionality mismatch
- **IN GENERAL:**
  - Is the model not learning anything at all?

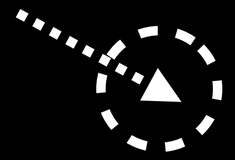


What this means for us

- Deep learning algorithms (Machine Learning in general) are susceptible to manipulative attacks
    - Use with caution in critical deployments
- Don’t make false assumptions about what/how the model learns
- Evaluate a model’s **adversarial resilience** - not just accuracy/precision/recall
- Spend effort to make models more robust to tampering

Defending the machines

- Distillation
  - Train model 2x, feed first DNN output logits into second DNN input layer
- Train model with adversarial samples
  - i.e. ironing out imperfect knowledge learnt in the model
- Other miscellaneous tweaks
  - Special regularization/loss-function methods (simulating adversarial content during training)



DEEP-PWNING

“metasploit for machine learning”

WHY DEEP-PWNING?

- **lol why not**
- “Penetration testing” of statistical/machine learning systems
- Train models with adversarial samples for increased robustness

PLEASE PLAY WITH IT &

CONTRIBUTE!

Deep Learning and Privacy

- **Deep learning also sees challenges in other areas relating to security & privacy**
- **Adversary can reconstruct training samples from a trained black box DNN model** (Fredrikson, 2015)
- **Can we precisely control the learning objective of a DNN model?**
- **Can we train a DNN model without the training agent having complete access to all training data?** (Shokri, 2015)





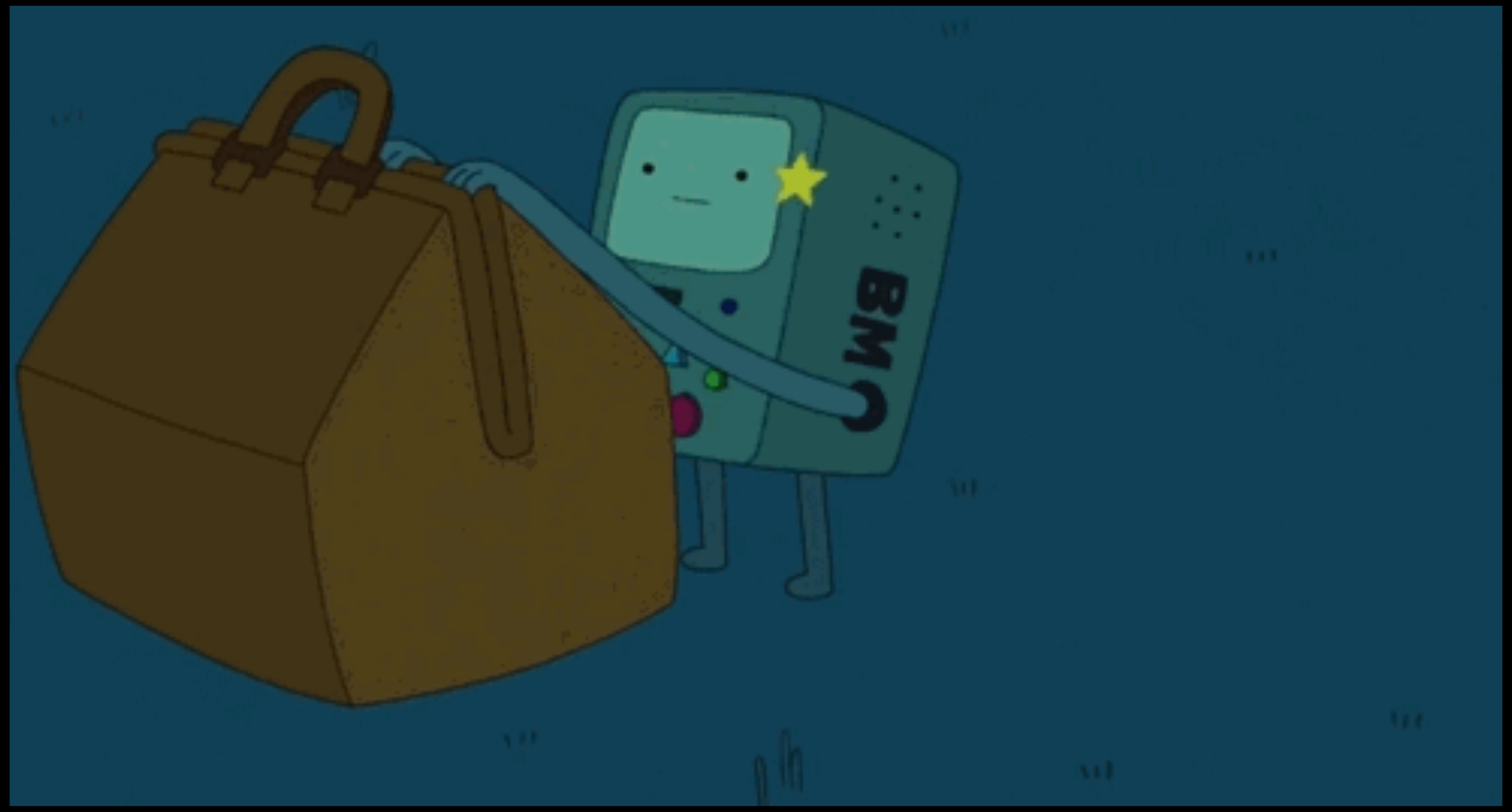
WHY IS THIS IMPORTANT?

WHY DEEP-PWNING?

- MORE CRITICAL SYSTEMS RELY ON MACHINE LEARNING → MORE IMPORTANCE ON ENSURING THEIR ROBUSTNESS

- WE NEED PEOPLE WITH BOTH SECURITY AND STATISTICAL SKILL SETS TO DEVELOP ROBUST SYSTEMS AND EVALUATE NEW INFRASTRUCTURE

LEARN IT OR BECOME IRRELEVANT



Data Mining for Cyber Security (SF Bay Area)

Home | Members | Sponsors | Photos | Pages | Discussions | More | Group tools | My profile



Change photo

San Francisco, CA
Founded Apr 30, 2015

About us...

Invite friends

Members 1,572

Welcome!

+ Schedule a new Meetup

Upcoming **Past** Calendar

April 12 · 6:30 PM

DMCS Meetup #8 - Measuring User Authenticity and SaaS Security Product Building



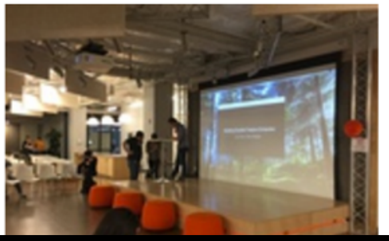
112 Members | ☆☆☆☆☆ | 2 Photos

RSVP EARLY - SPACE IS LIMITED We were on a couple months break since many companies were busy at RSA earlier in March. This coming meetup will be held at GSVlabs,... LEARN MORE

What's new







@cchio

MLHACKER