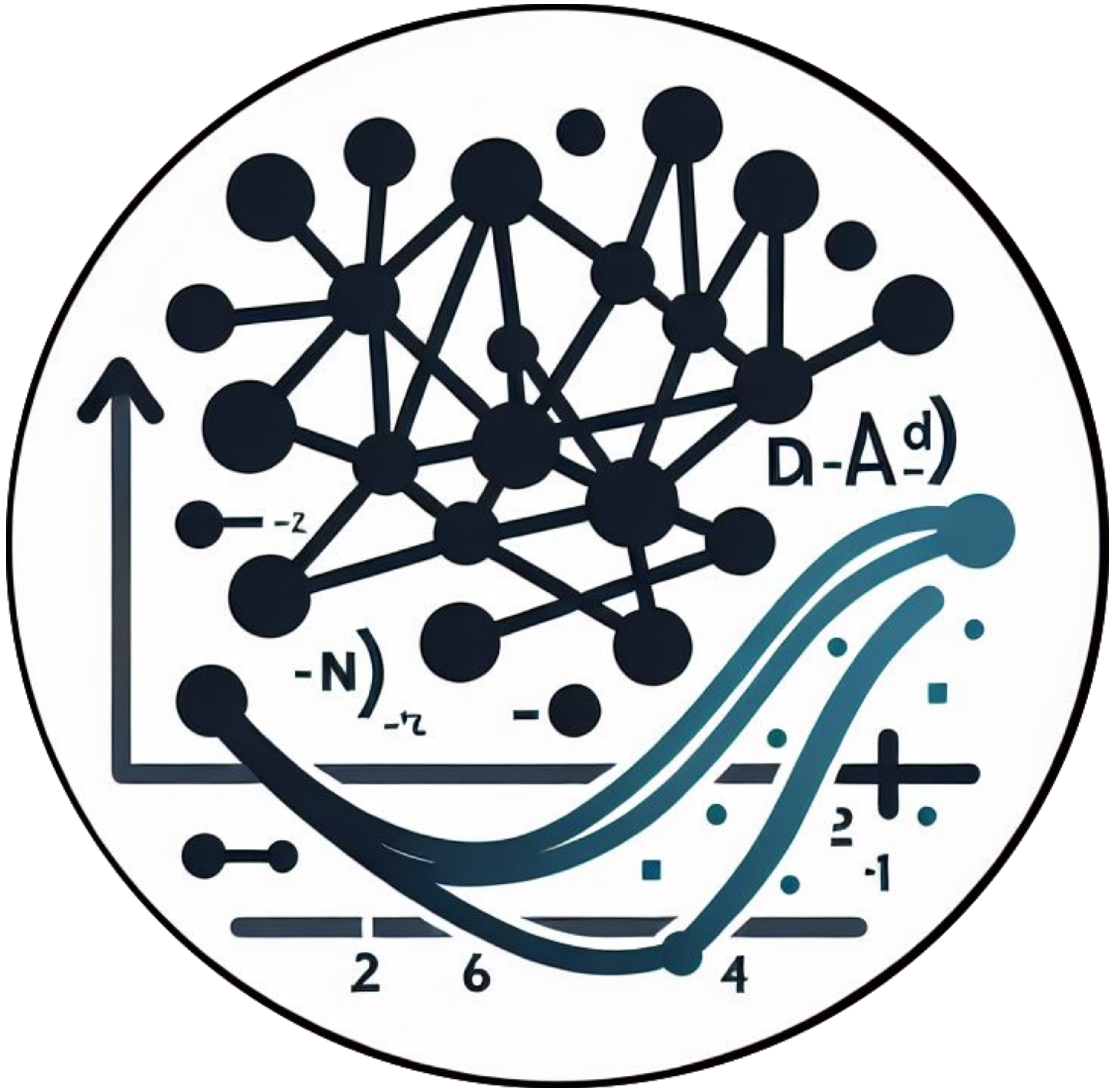
Regularization

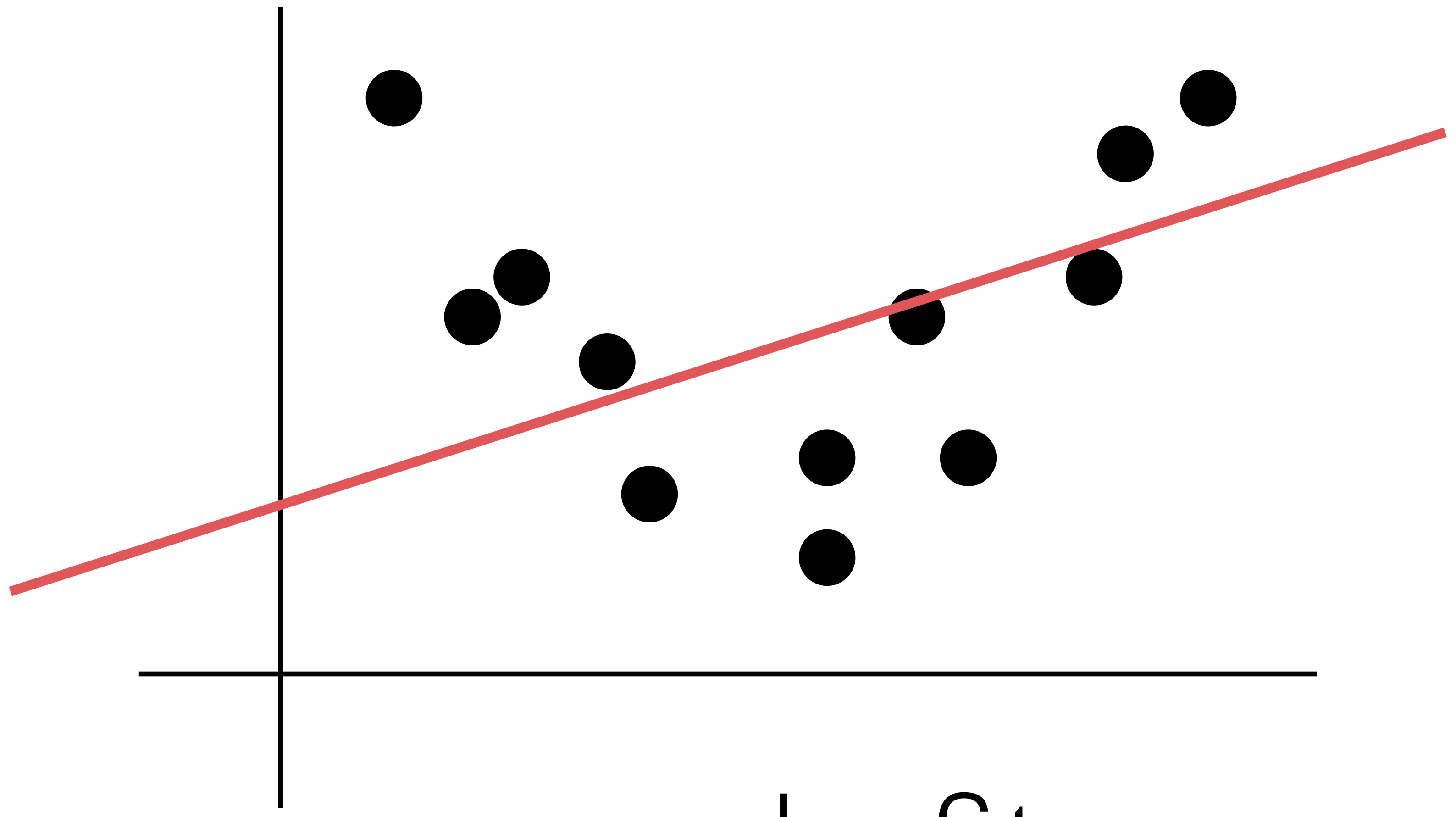


Deep Learning for Engineers

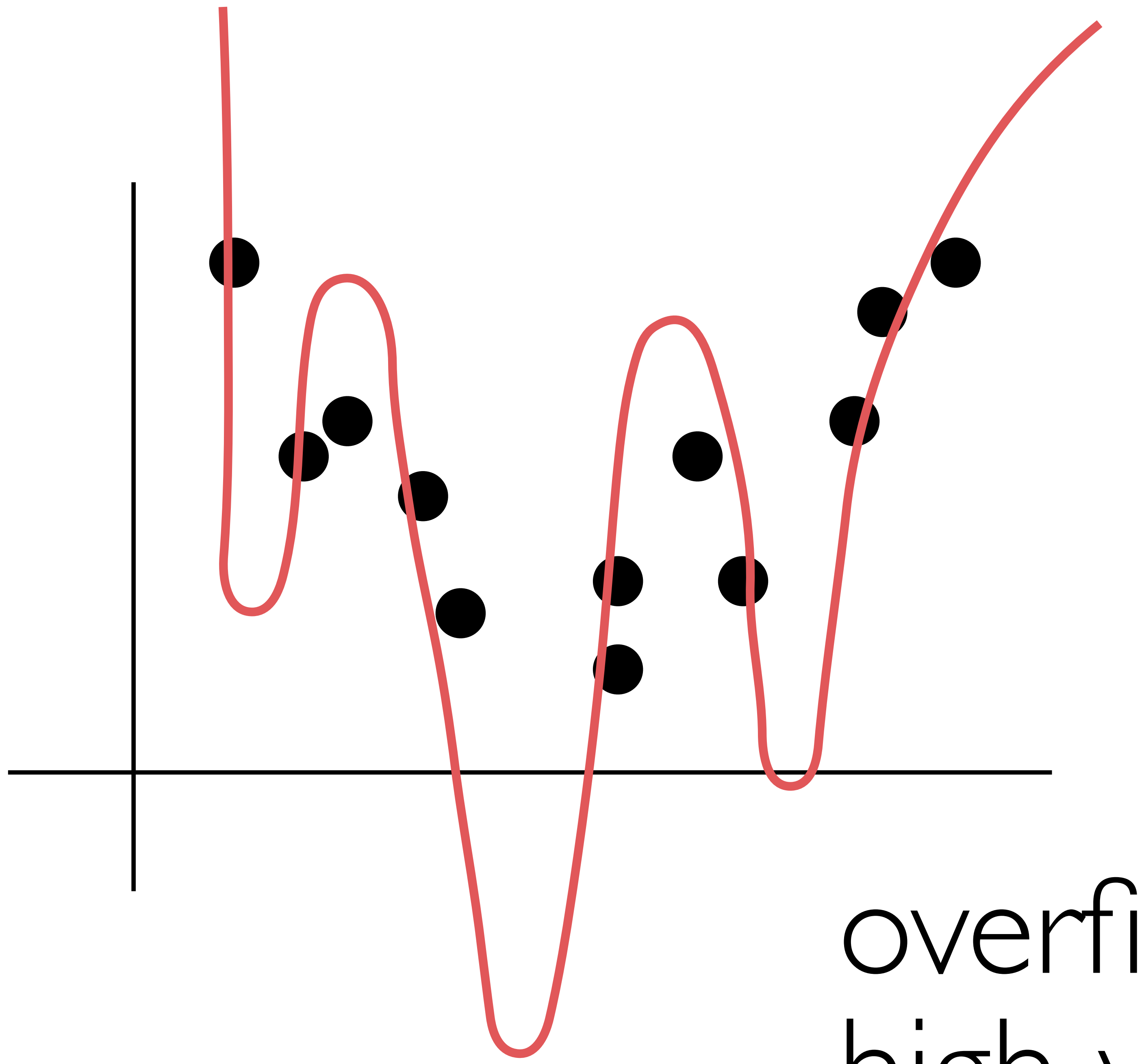
Andrew Ning
aning@byu.edu

Midterm

Validation (underfitting vs overfitting or bias and variance)

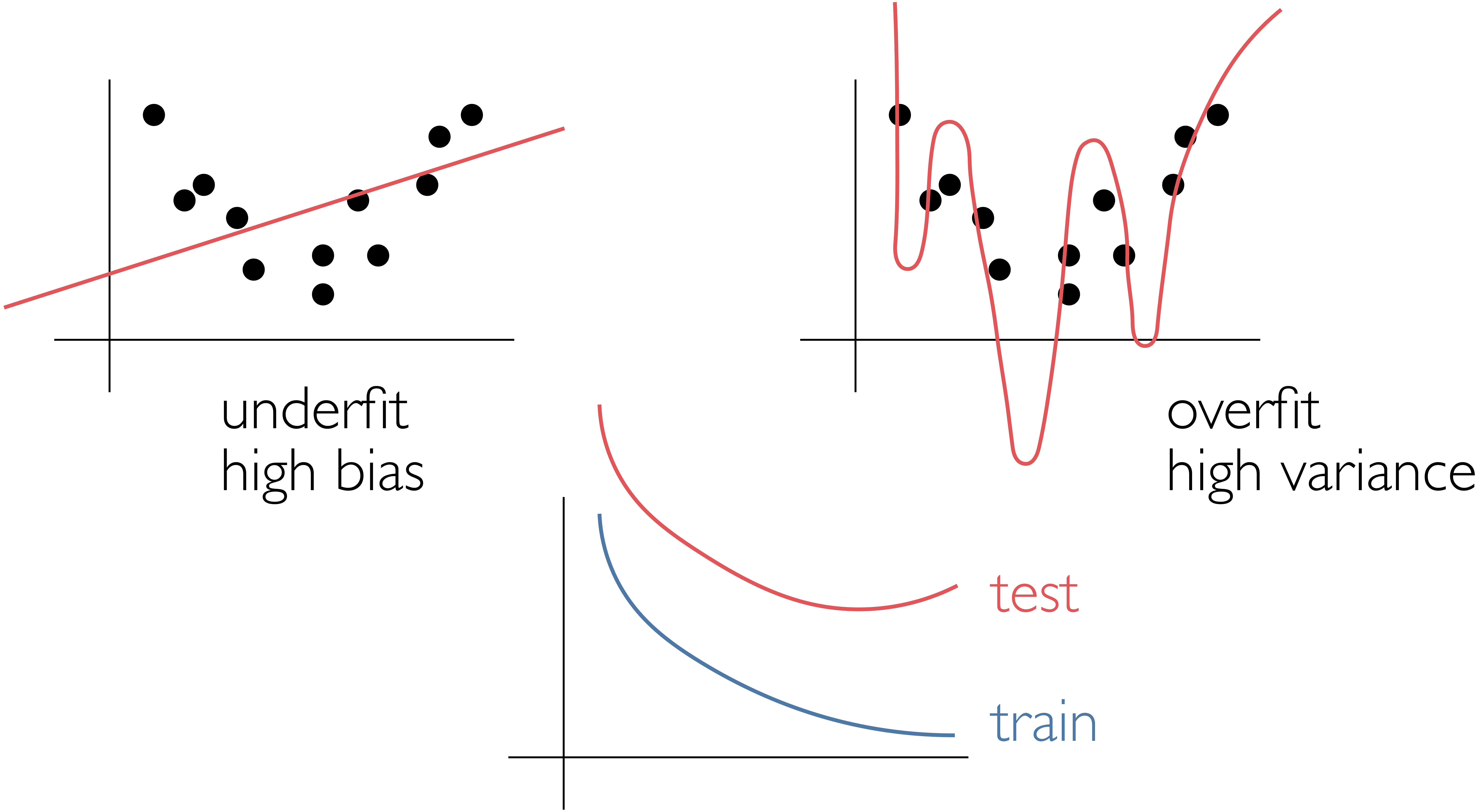


underfit
high bias



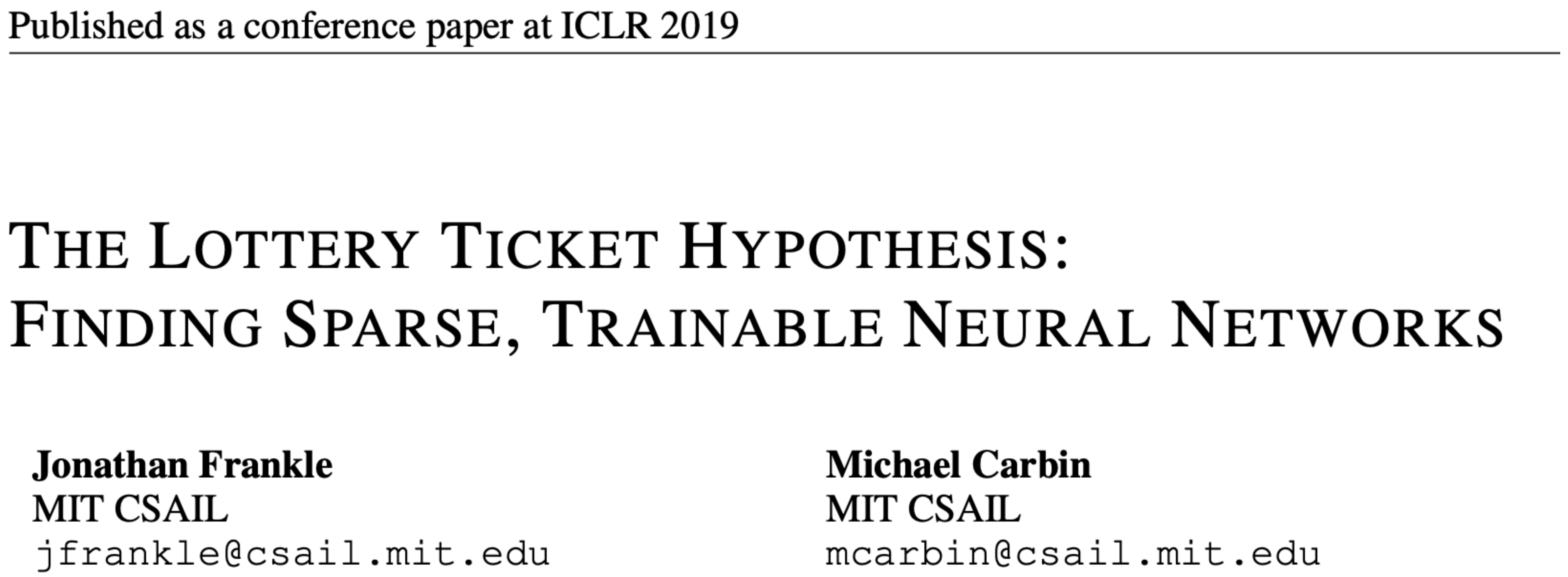
overfit
high variance

Validation (underfitting vs overfitting or bias and variance)



First address underfitting (high bias)

Simpler Network



Published as a conference paper at ICLR 2019

THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

brainstorm: what else can I do for overfitting?

Get more data (different data)

and/or augment data

Add more physics

Early Stopping

Weight regularization

$$\frac{\gamma}{2}||W||_2^2$$

Weight regularization

$$\frac{\gamma}{2}\|W\|_2^2$$

decoupled weight decay

```
torch.optim.AdamW(model.parameters(), lr=1e-4,
    weight_decay=1e-5)
```

```
torch.optim.Adam(model.parameters(), lr=1e-4,
    weight_decay=1e-5, decoupled_weight_decay=True)
```

gradient descent

$$W^{(k+1)} = W^{(k)} - \alpha \nabla L(W)$$

gradient descent

$$W^{(k+1)} = W^{(k)} - \alpha \nabla L(W)$$

L2 regularization

$$L = L_{data} + \frac{\gamma}{2} \|W\|_2^2$$

$$\nabla L = \nabla L_{data} + \gamma W$$

$$W^{(k+1)} = W^{(k)} - \alpha \nabla L_{data} \boxed{- \alpha \gamma W^{(k)}}$$

only showing weights (one generally doesn't decay biases)

Decoupled Weight Decay Regularization

Ilya Loshchilov & Frank Hutter
University of Freiburg
Freiburg, Germany,
{ilya,fh}@cs.uni-freiburg.de

RMSProp (part of Adam for simplicity)

$$W^{(k+1)} = W^{(k)} - \frac{\alpha}{\sqrt{s}} \nabla L(W)$$

RMSProp (part of Adam for simplicity)

$$W^{(k+1)} = W^{(k)} - \frac{\alpha}{\sqrt{s}} \nabla L(W)$$

L2 regularization

$$\nabla L = \nabla L_{data} + \gamma W$$

$$W^{(k+1)} = W^{(k)} - \frac{\alpha}{\sqrt{s}} \nabla L_{data} \boxed{- \frac{\alpha}{\sqrt{s}} \gamma W^{(k)}}$$

not what we want. weight decay is scaled differently for each parameter. and changes each iteration

RMSProp (part of Adam for simplicity)

$$W^{(k+1)} = W^{(k)} - \frac{\alpha}{\sqrt{s}} \nabla L(W)$$

L2 regularization

$$\nabla L = \nabla L_{data} + \gamma W$$

$$W^{(k+1)} = W^{(k)} - \frac{\alpha}{\sqrt{s}} \nabla L_{data} \boxed{- \frac{\alpha}{\sqrt{s}} \gamma W^{(k)}}$$

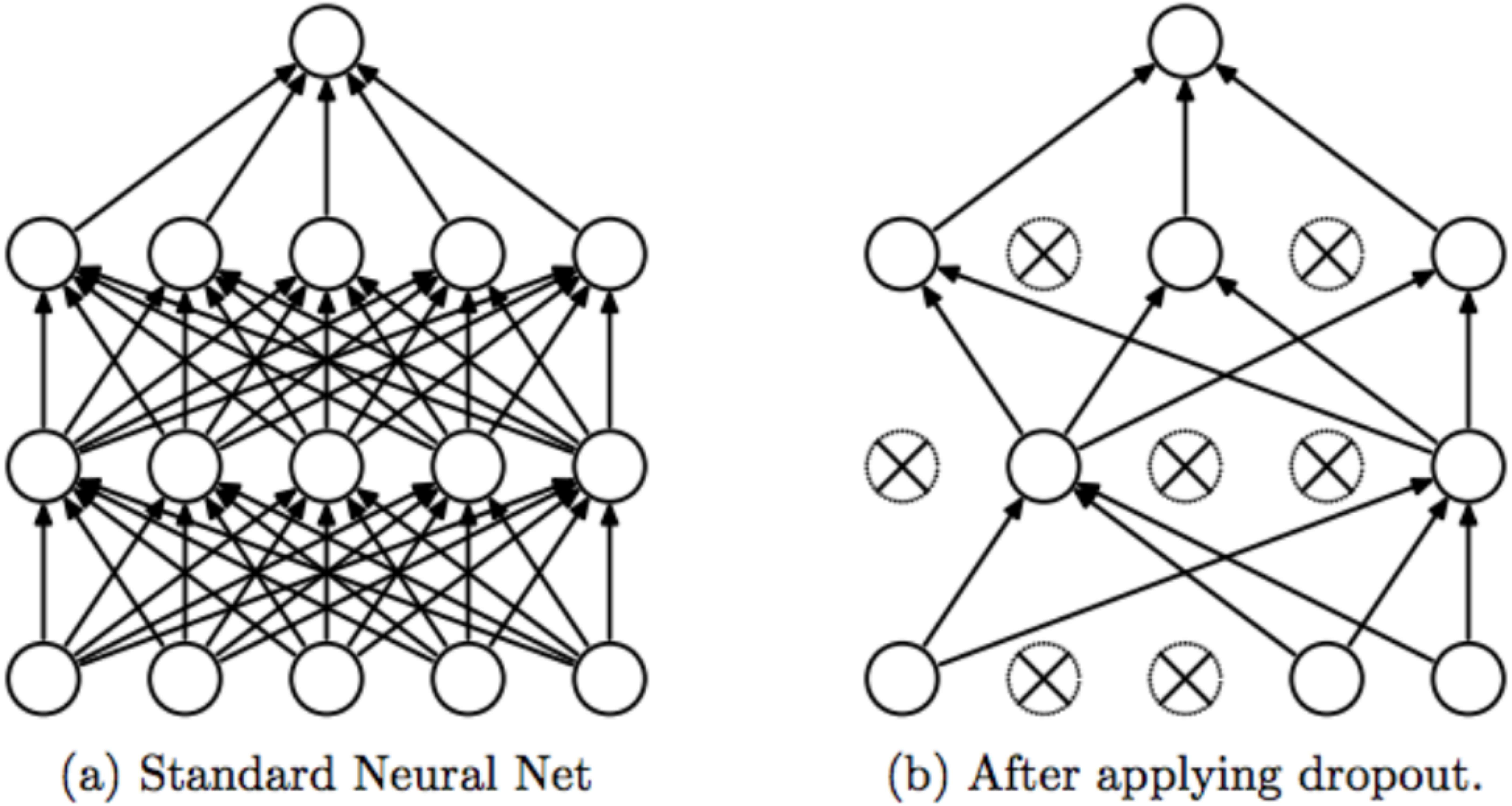
not what we want. weight decay is scaled differently for each parameter. and changes each iteration

AdamW (decoupled weight decay)

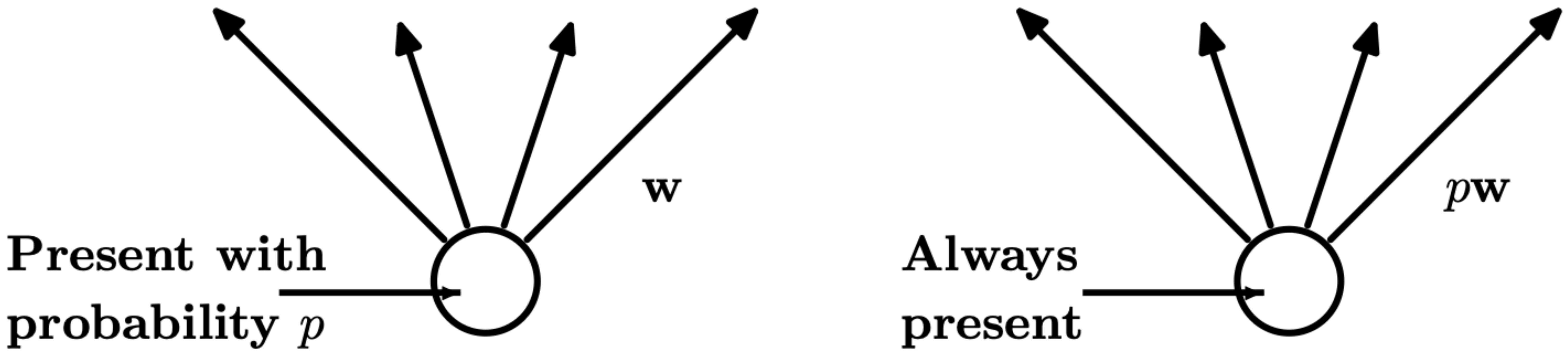
$$W^{(k+1)} = W^{(k)} - \frac{\alpha}{\sqrt{s}} \nabla L_{data} \boxed{- \alpha \gamma W^{(k)}}$$

Ensemble averaging

Dropout



Dropout: A Simple Way to Prevent Neural Networks from Overfitting.
Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov
Journal of Machine Learning Research 15 (2014) 1929-1958



(a) At training time

(b) At test time

nn.Dropout(p=0.2)

(insert after activation functions)