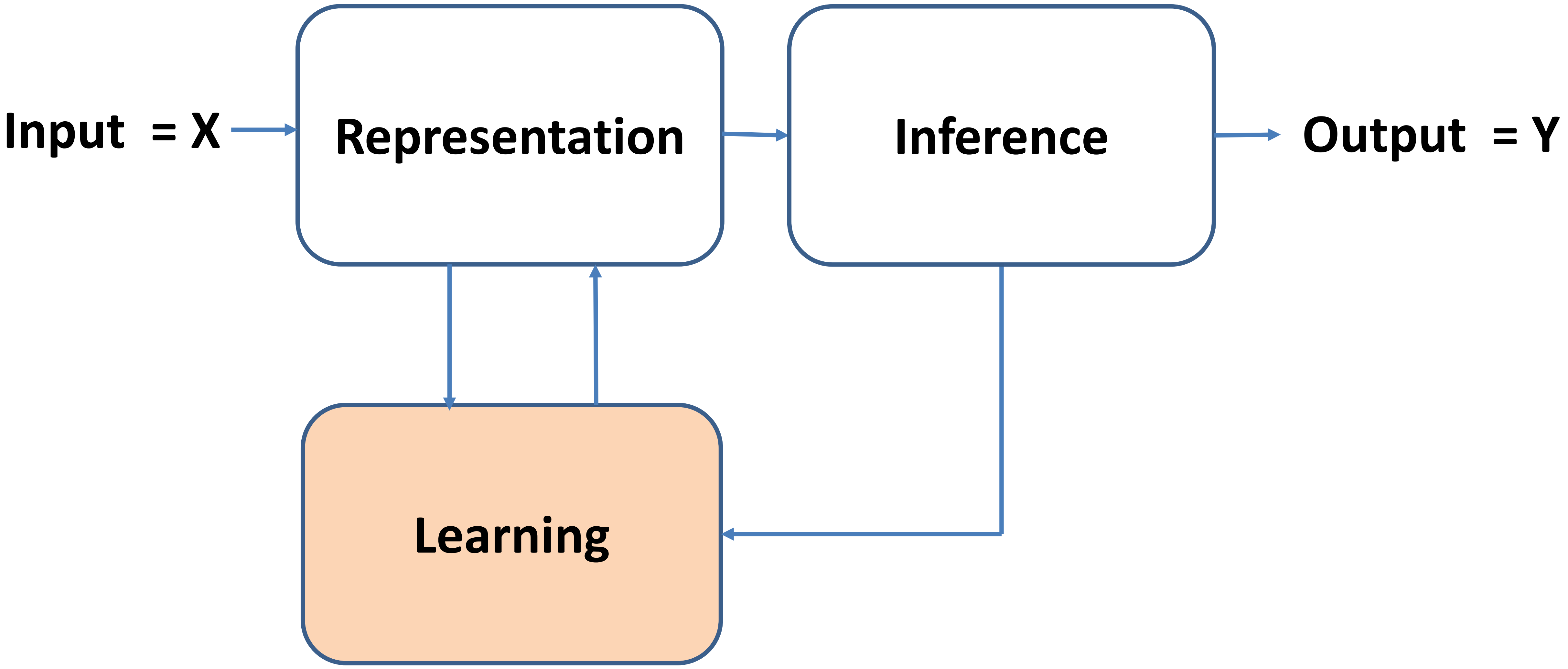
04 | Regularization

Stephen F Elston | Principle Consultant, Quantia Analytics, LLC

Microsoft

Introduction to Regularization for Deep Learning



Introduction to Regularization for Deep Learning

- Deep learning models have very large numbers of parameters which must be learned.
  - Even with large training datasets there may only be a few samples per parameters
- Large number of parameters leads to high chance of **over-fitting** deep learning models
  - Over-fit models do not generalize
  - Over-fit models have poor response to input noise
- To prevent over-fitting we apply **regularization methods**

Introduction to Regularization for Deep Learning

- Bias-variance trade-off
- l2 regularization
- l1 regularization
- Early stopping
- Dropout regularization
- Batch normalization

The Bias-Variance Trade-Off

- High capacity models fit training data well
  - Exhibit high variance
  - Do not generalize well; exhibit **brittle behavior**
  - $\text{Error}_{\text{training}} << \text{Error}_{\text{test}}$
- Low capacity models have high bias
  - Generalize well
  - Do not fit data well

- Regularization adds bias
  - Strong regularization adds significant bias
  - Weak regularization leads to high variance

The Bias-Variance Trade-Off

- How can we understand the bias-variance trade-off?
- We start with the error:

$$\Delta y = E\left[Y - \hat{f}(X)\right]$$

Where:

$Y =$ the label vector.

$X =$ the feature matrix.

$\hat{f}(x) =$ the trained model.

The Bias-Variance Trade-Off

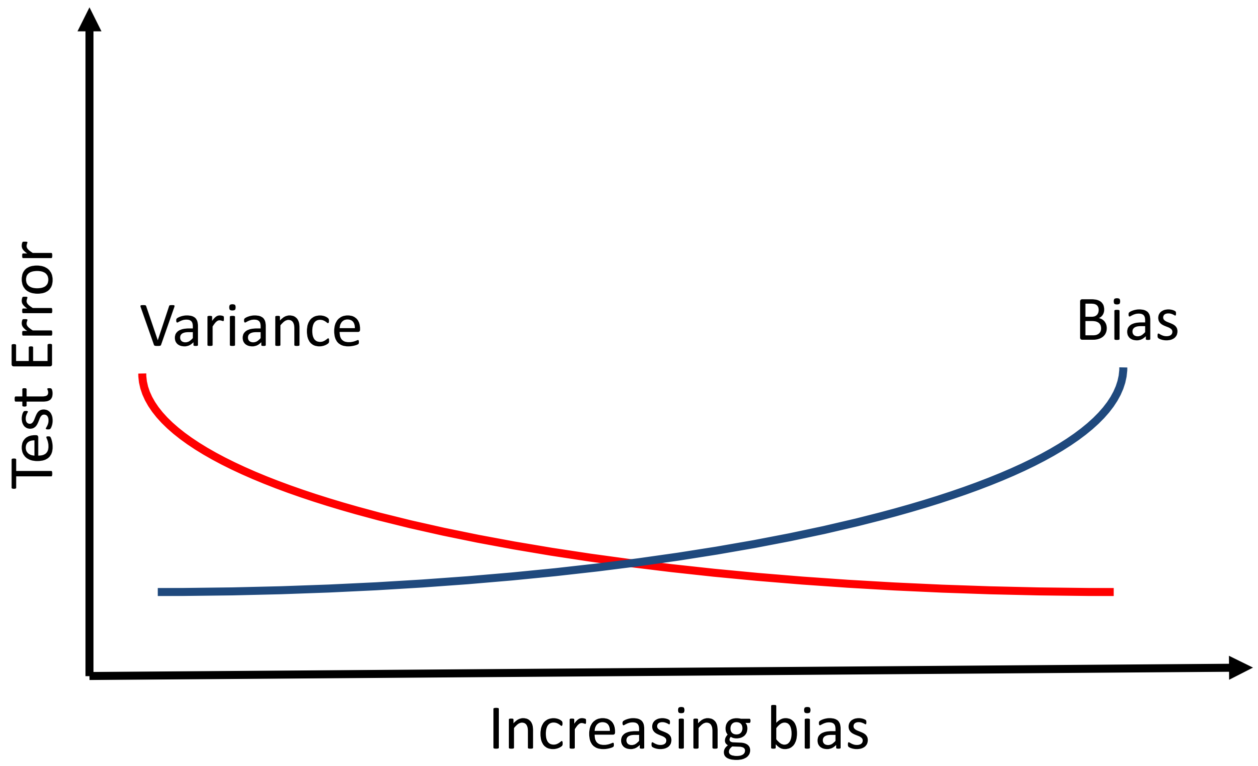
- We can expand the error term

$$\Delta x = \left(E[\hat{f}(X)] - \hat{f}(X)\right)^2 + E\left[(\hat{f}(X) - E[\hat{f}(X)])^2\right] + \sigma^2$$

$$\Delta x = Bias^2 + Variance + Irreducible\ Error$$

- Increasing bias decreases variance
- Notice that even if the bias and variance are 0 there is still irreducible error

The Bias-Variance Trade-Off



l2 Regularization

- Over-fit models tend to have parameters (weights) with extreme values
- One way to regularize models is to limit the values of the parameters

l2 Regularization

- One way to limit the size of the model parameters is to constrain the **l2** or **Euclidian norm**:

$$||W||^2 = \left(w_1^2 + w_2^2 + \ldots + w_n^2\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} w_i^2\right)^{\frac{1}{2}}$$

- The regularized loss function is then:

$$J(W) = J_{MLE}(W) + \lambda ||W||^2$$

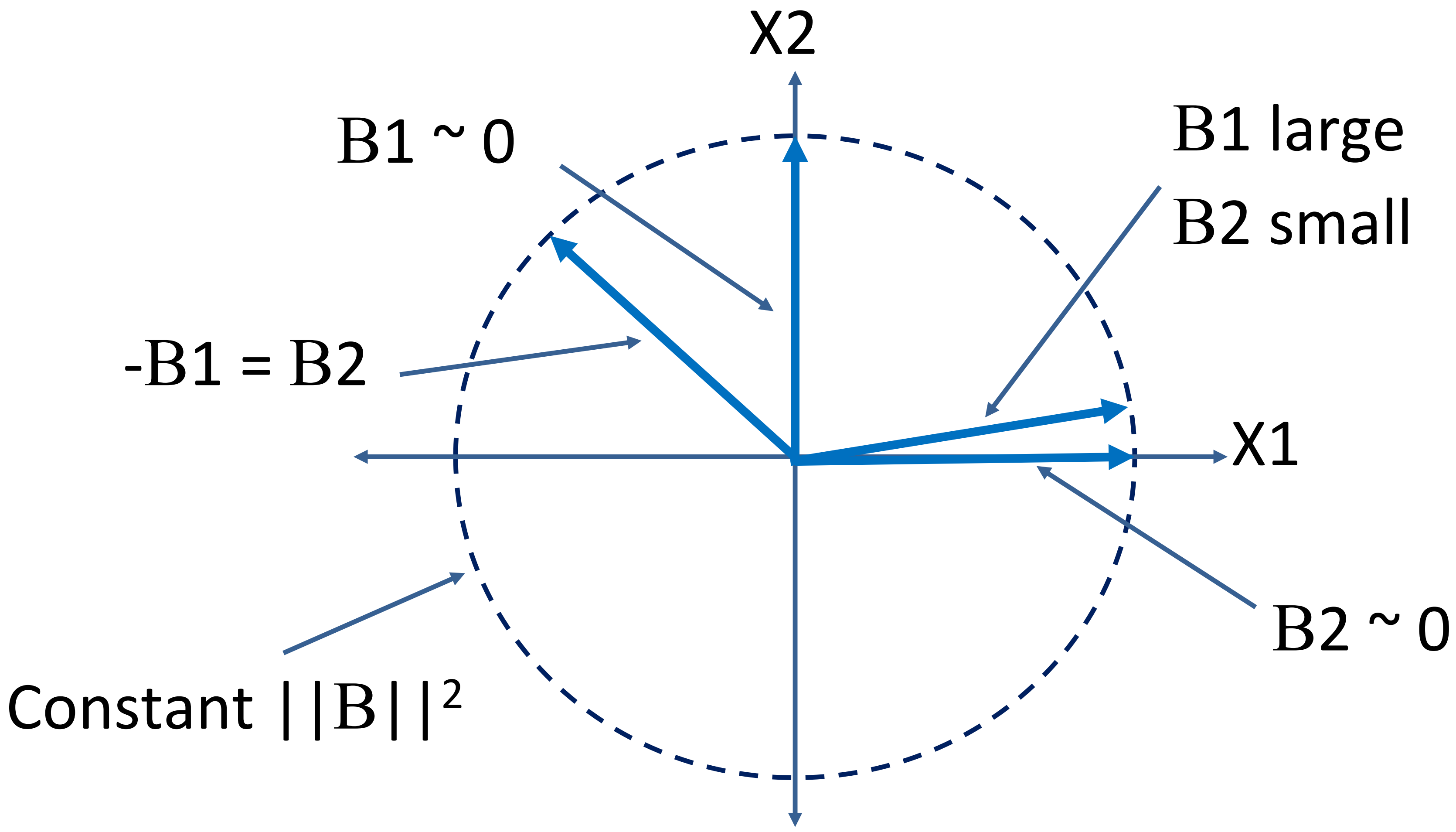
- Where λ is the regularization hyperparameter
  - Large λ increases bias but reduces variance
  - Small λ decreases bias and increases variance

l2 Regularization

- l2 regularization goes by many names
- Is called Euclidian norm regularization
- First published by Andrey Tikhonov regularization, in late 1940s
  - Only published in English in 1977
  - Is known as **Tikhonov regularization**
- In the statistics literature the method is often called **ridge regression**
- In the engineering literature is referred to as **pre-whitening**

l2 Regularization

How can you gain some intuition about l2 regularization?



l2 regularization is considered a **soft constraint**

l1 Regularization

- Regularization can be performed with other norms
- The **l1 (min-max) norm** is another common choice
- Conceptually, l1 norm limits the sum of the absolute values of the weights:

$$||W||^1 = \left(|w_1| + |w_2| + \ldots + |w_n|\right) = \left(\sum_{i=1}^{n} |w_i|\right)^1$$

- The l1 norm is also known as the **Manhattan distance** or **taxi cab distance**, since it is the distance traveled on a grid between two points.

l1 Regularization

- Given the l1 norm of the weights, the loss function becomes:

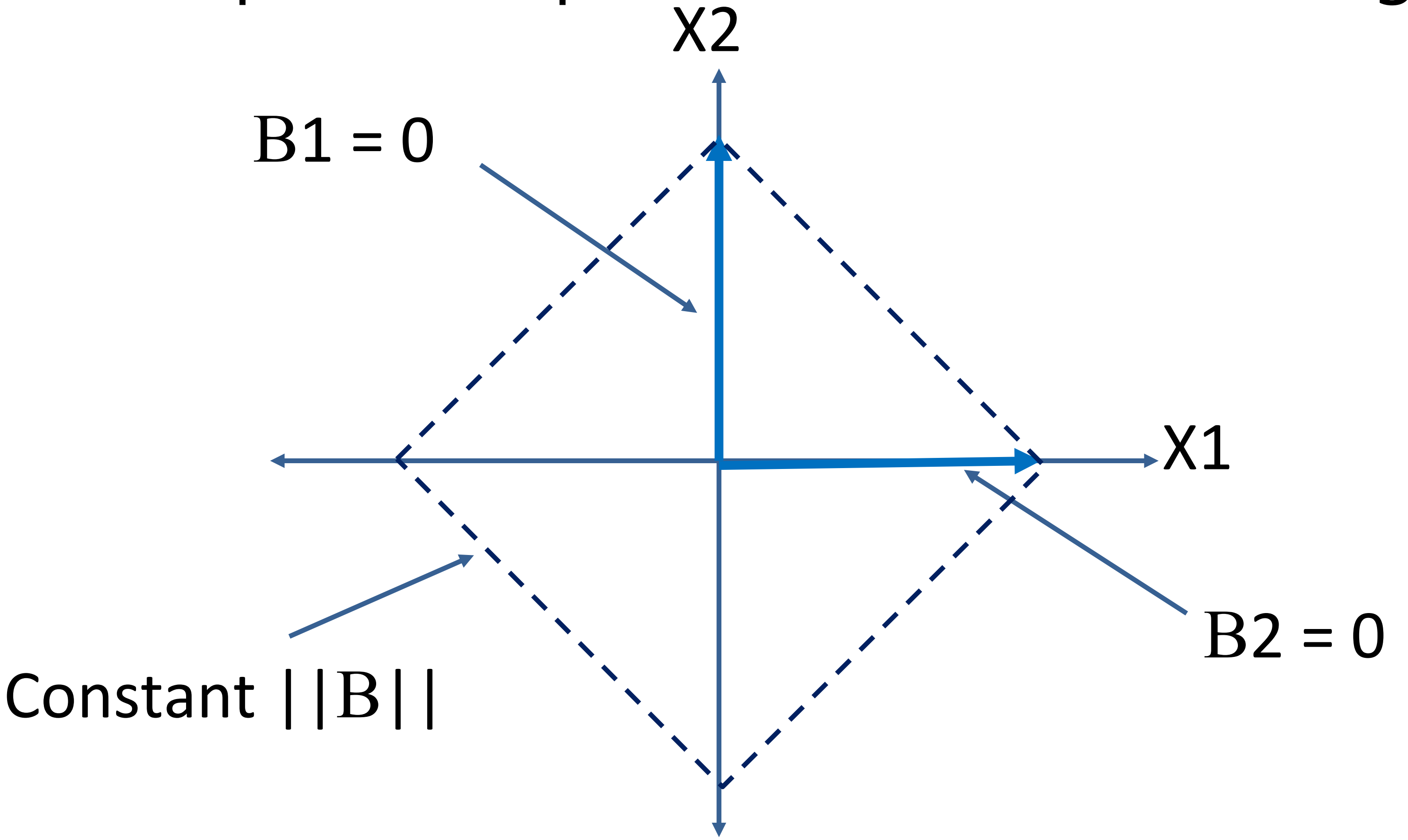
$$J(W) = J_{MLE}(W) + \alpha ||W||^1$$

- Where α is the regularization hyperparameter
  - Large α increases bias but reduces variance
  - Small α decreases bias and increases variance

- The l1 constraint drives some weights to exactly 0
  - This behavior leads to the term **lasso regularization**

l1 Regularization

A diagram helps develop some intuition on l1 regularization:



L1 regularization is a **hard constraint** on the weights

Early Stopping

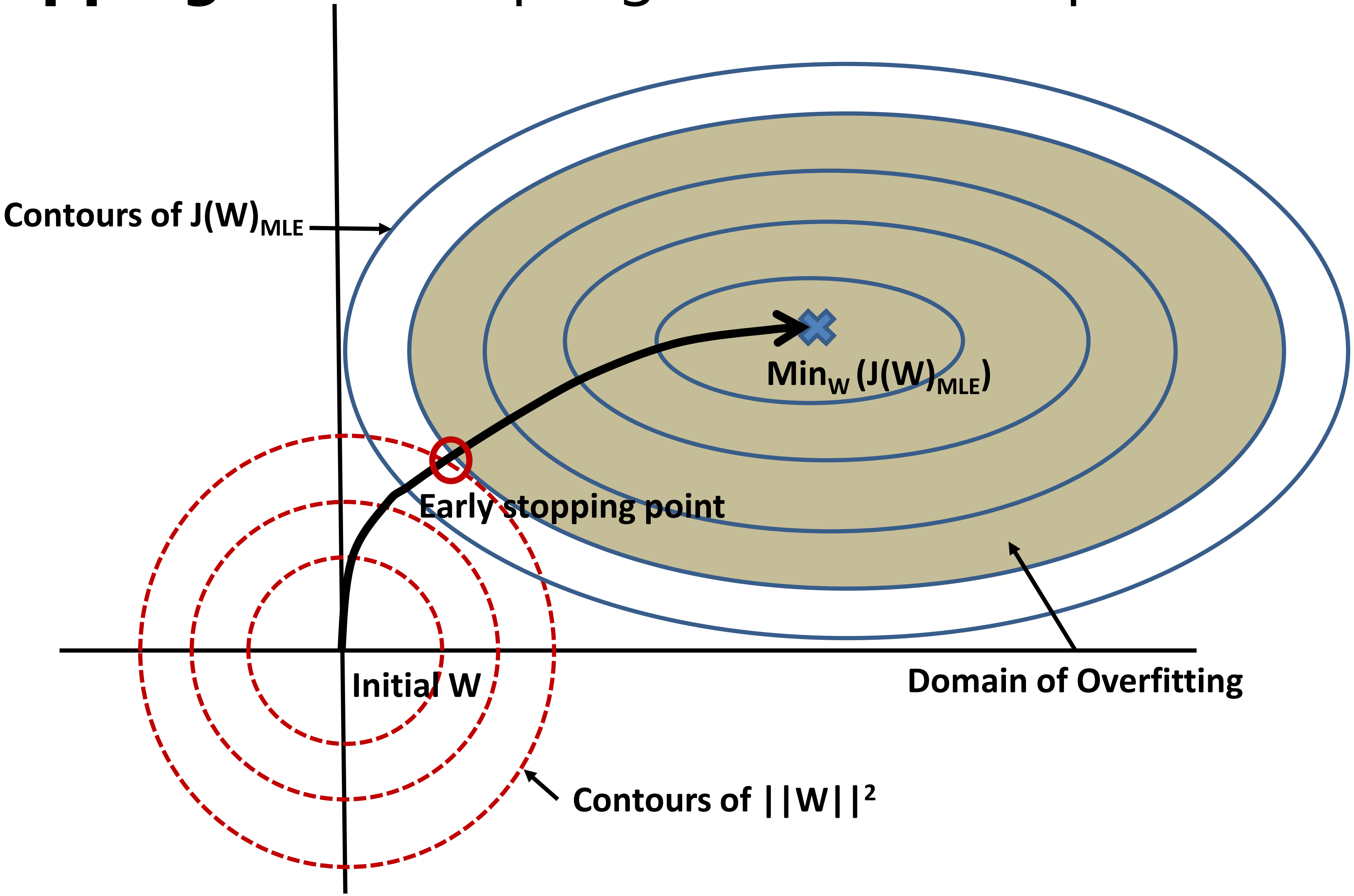
- **Early stopping** is an old and simple idea
- Stop updating the model weights before the model becomes overfit
- Early stopping is analogous to l2 regularization
- We can formulate the regularized loss function as:

$$argmin_W J(W) = J(W)_{MLE} + \alpha ||W||^2$$

- Where α is the regularization hyperparameter

Early Stopping

Early stopping has a simple geometric interpretation



Dropout regularization

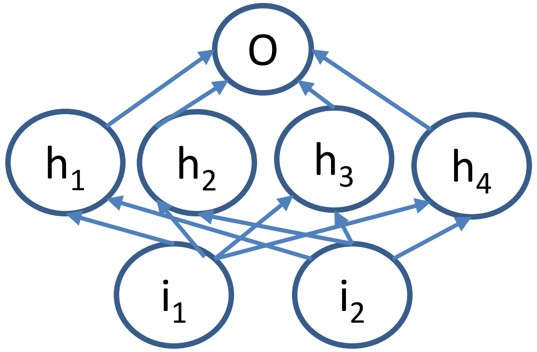
- Overfit deep network models tend to suffer from a problem of co-adaptation
  - With limited training data weight tensors become adapted to the training data
  - Such a model is unlikely to generalize
- We need a way to break the co-adaptation of the weight tensor

Dropout regularization

- **Dropout regularization** is a conceptually simple method unique to deep learning
  - At each step of the gradient decent some **fraction, p, of the weights are dropped-out** of each layer
  - The result is a series of models trained for each dropout sample
  - The final model is a **geometric mean** of the individual models
- Weight values are clipped in a small range as a further regularization

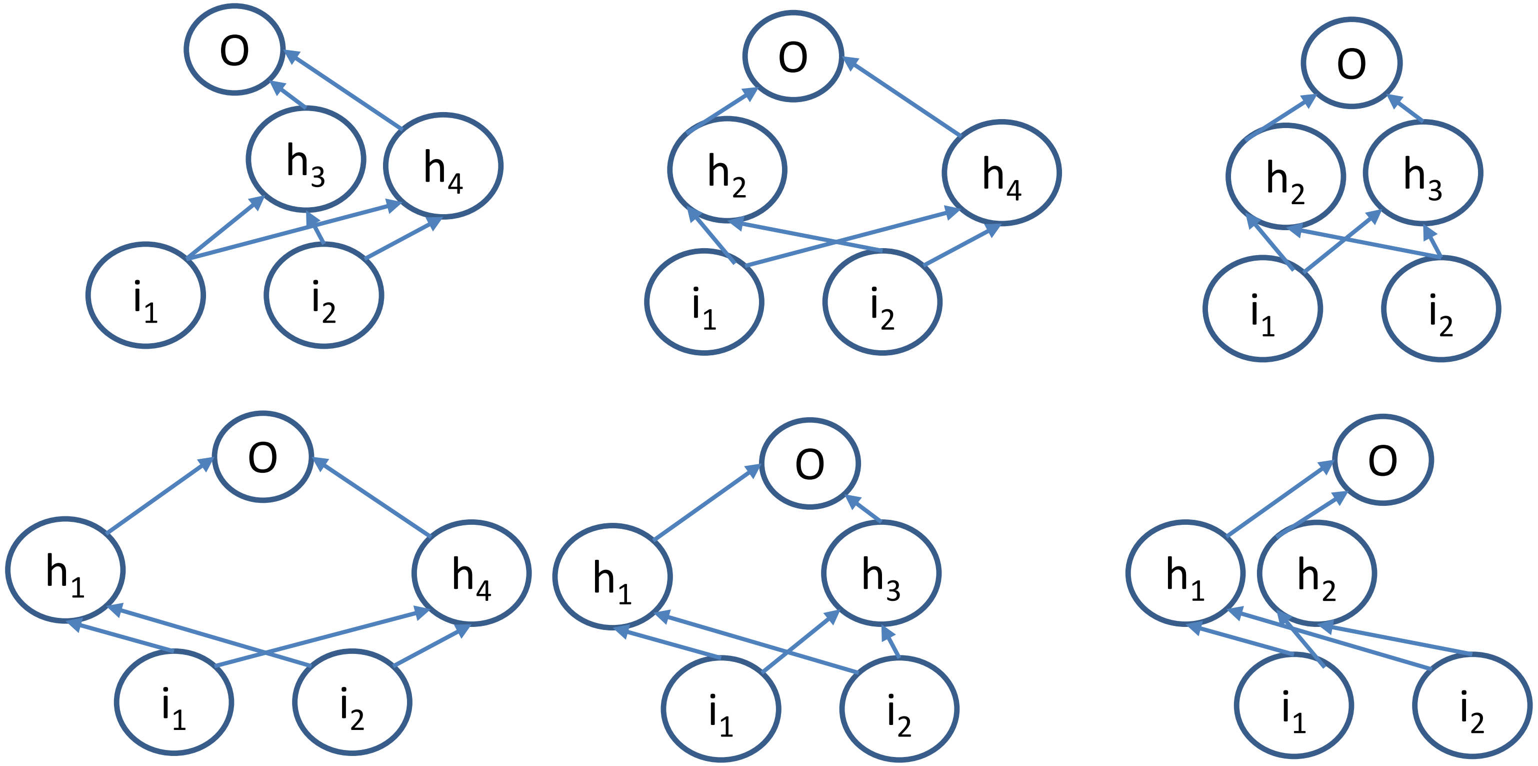
Dropout regularization

- Let's look at a simple example of a network with one hidden layer:



Dropout regularization

- For p = 0.5 here are six of the possible samples:



Batch Normalization

- In deep neural networks there is a high chance that units in a hidden layer have a **large range of output values**
  - Causes shifts in the covariance of the output values
  - Leads to difficulty computing the gradient
  - Slows convergence
- A solution is to normalize the output of the hidden layers in the network as a batch

Microsoft