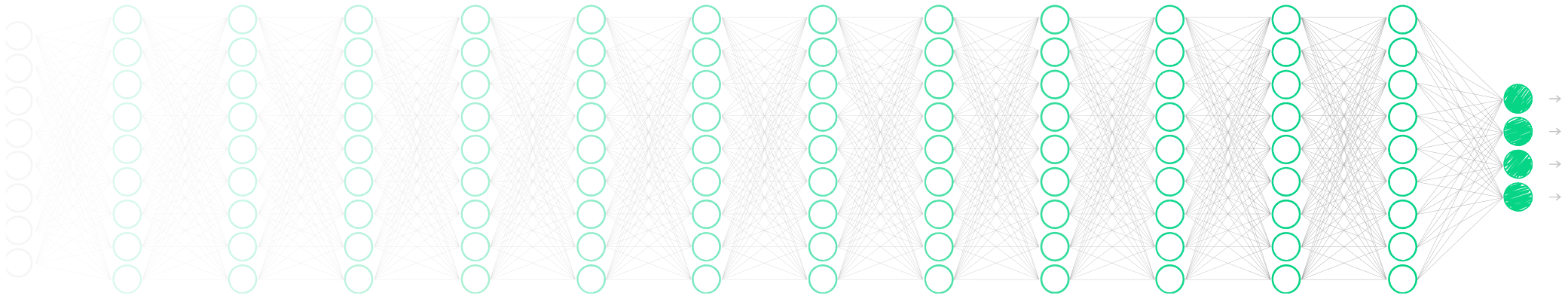
LECTURE 10-2

INITIALIZE WEIGHTS IN A SMART WAY

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml

Vanishing Gradient

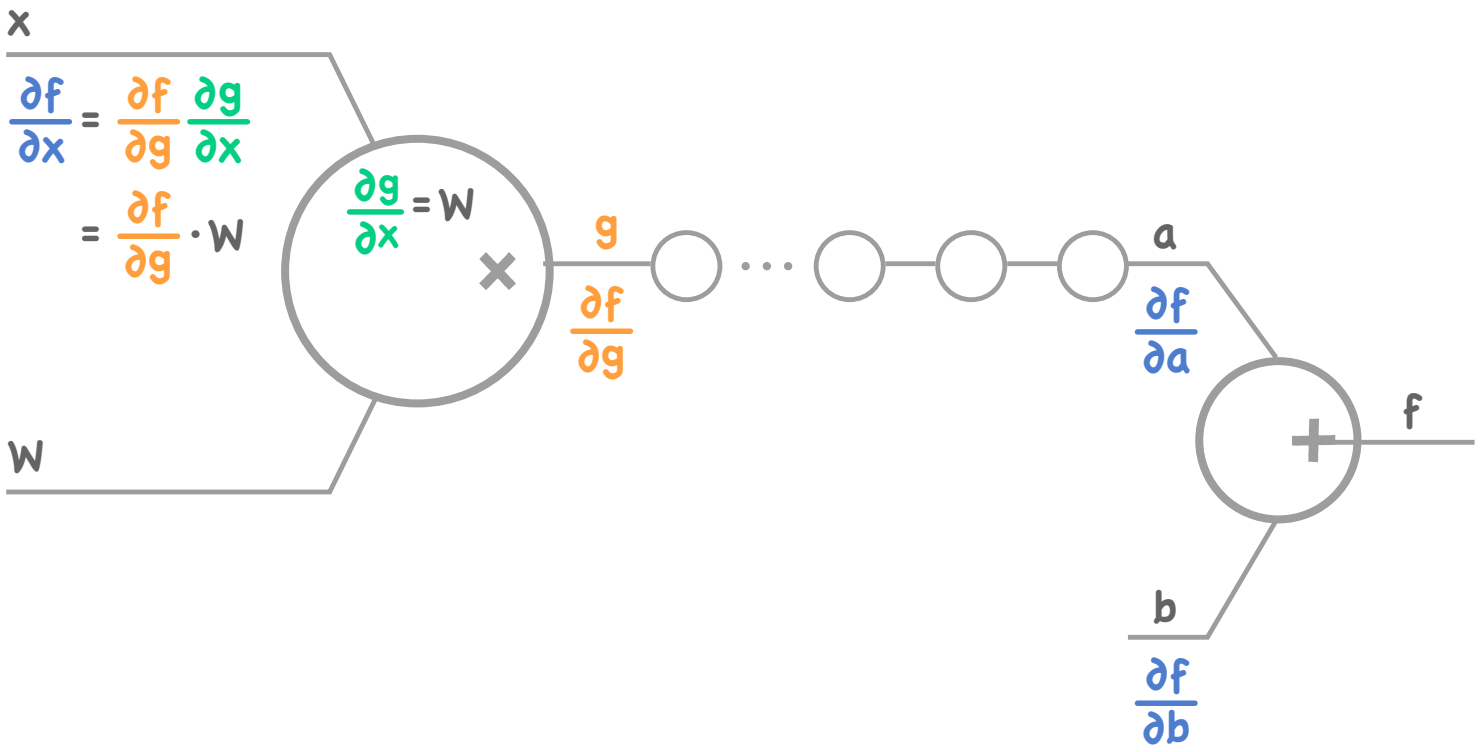


Geoffrey Hinton's Summary of Findings up to Today

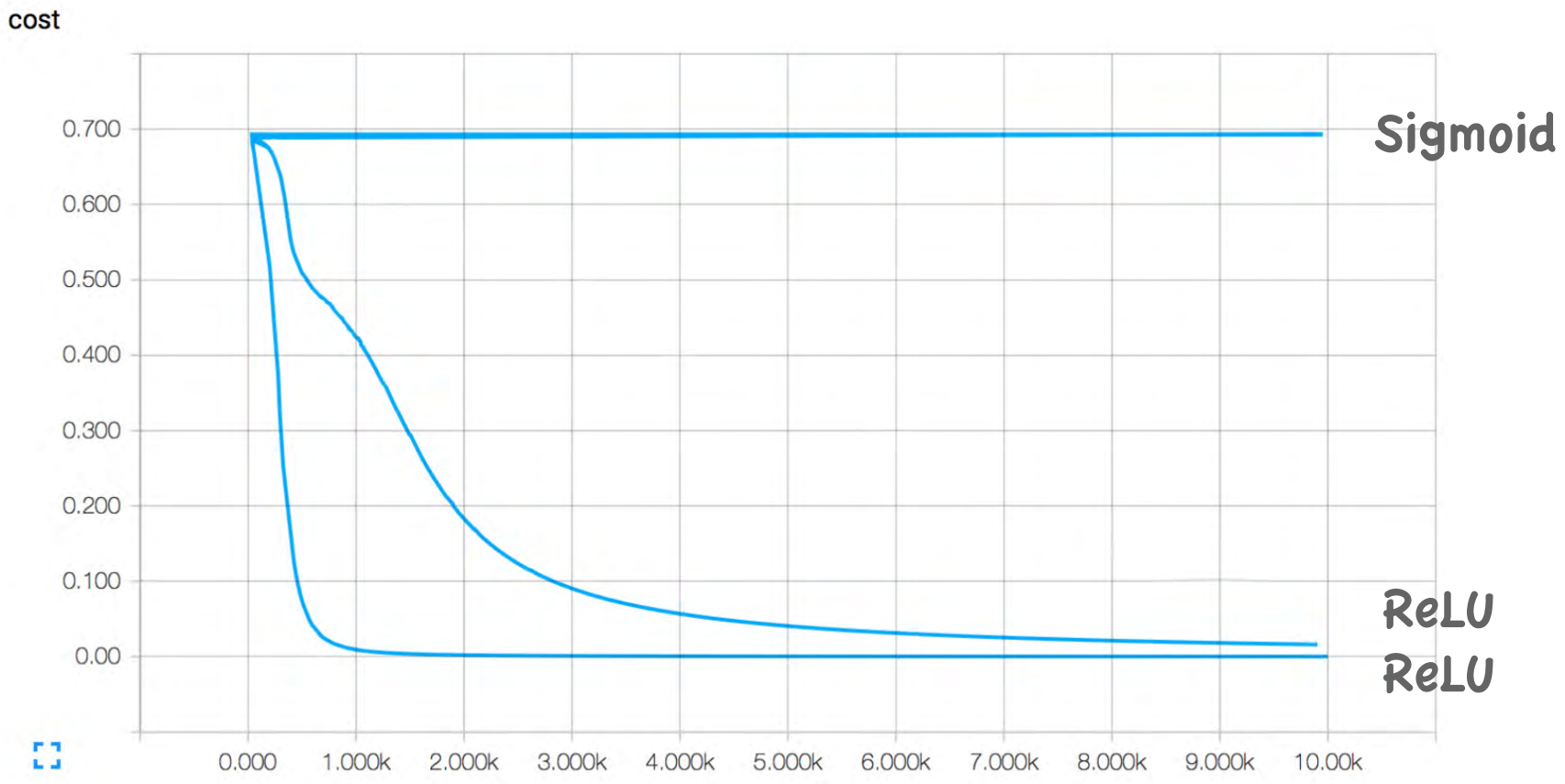
- Our labeled datasets were thousands of times too small
- Our computers were millions of times too slow
- We initialized the weights in a stupid way
- We used the wrong type of non-linearity

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-4/

Set All Initial Weights to 0



Cost Function

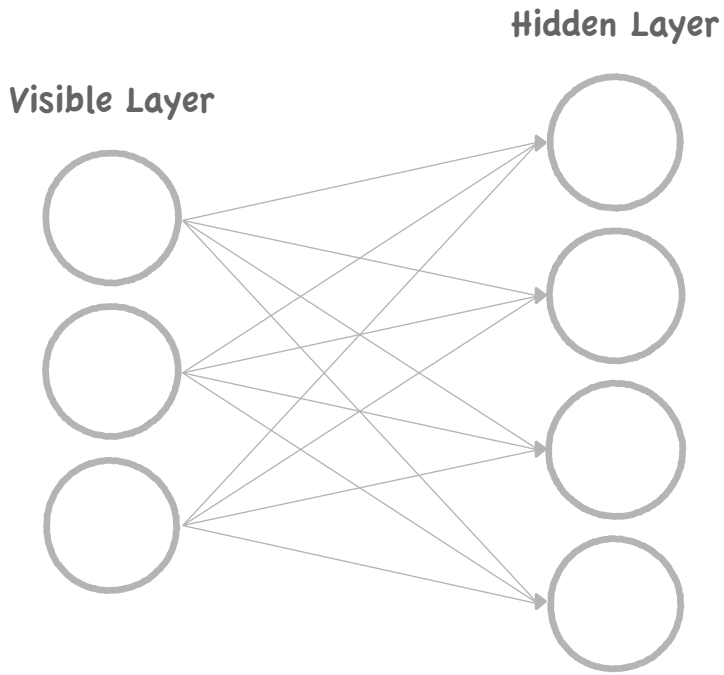


Need to Set the Initial Weight Values Wisely

- Not all 0's
- Challenging issue
- Hinton et al. (2006) "A Fast Learning Algorithm for Deep Belief Nets"
  - Restricted Boatman Machine (RBM)
- But still open problem

RBM Structure

Restriction = No Connections within a Layer

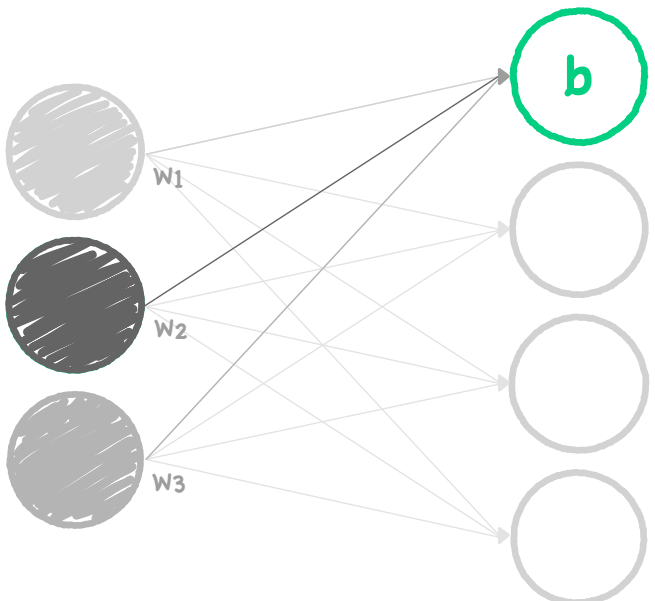


Recreate Input

(A) Forward

(B) Backward

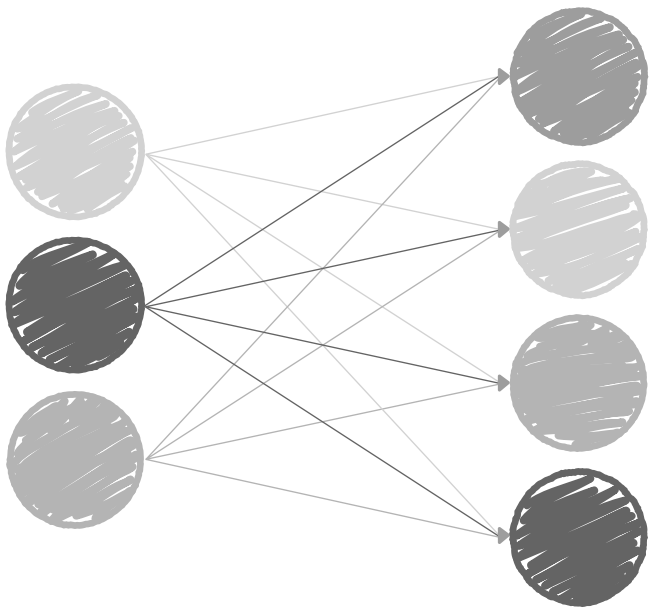
Recreate Input



(A) Forward

(B) Backward

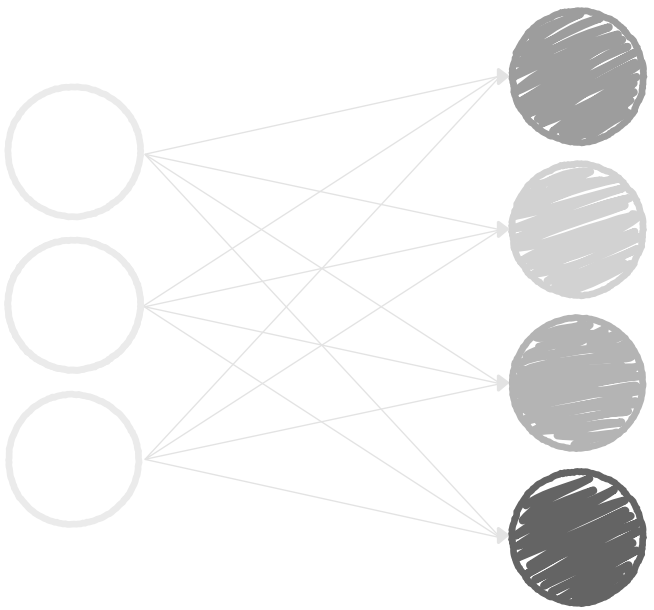
Recreate Input



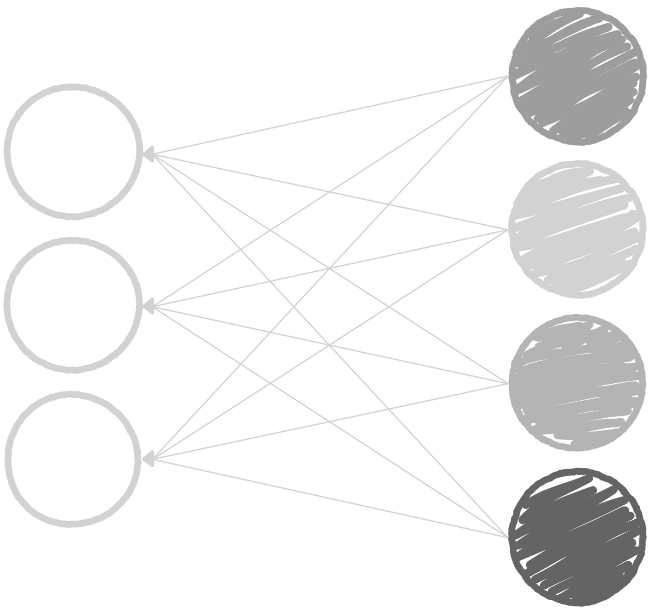
(A) Forward

(B) Backward

Recreate Input



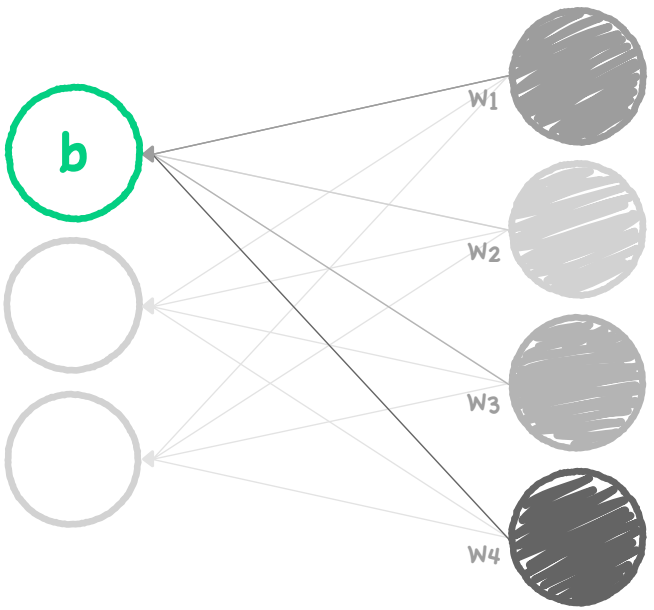
(A) Forward



(B) Backward

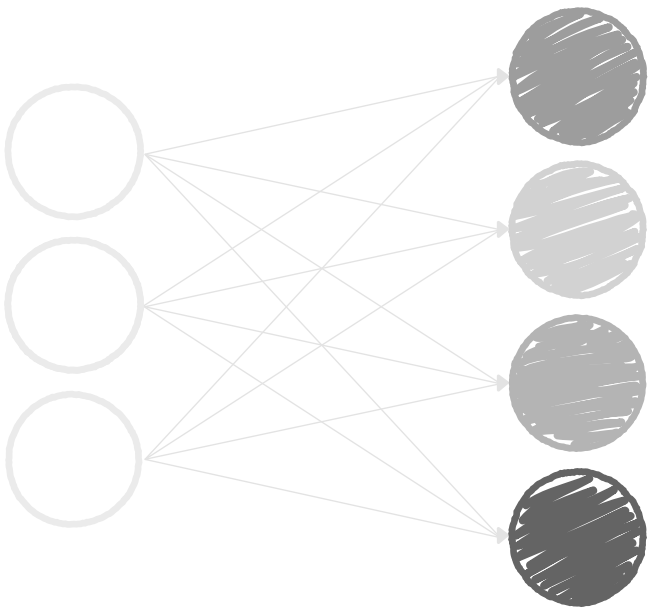
Recreate Input

(A) Forward

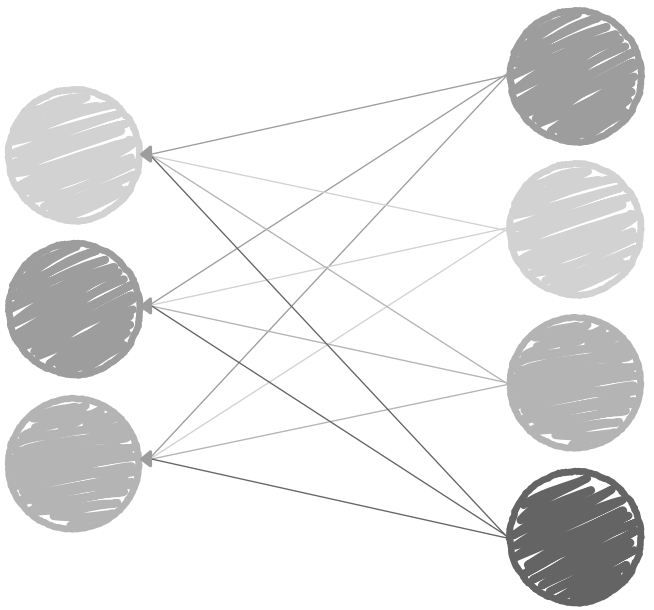


(B) Backward

Recreate Input

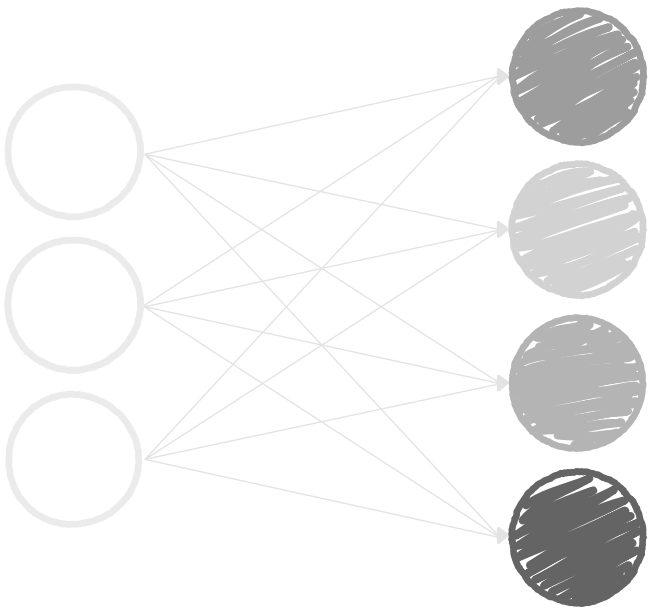


(A) Forward

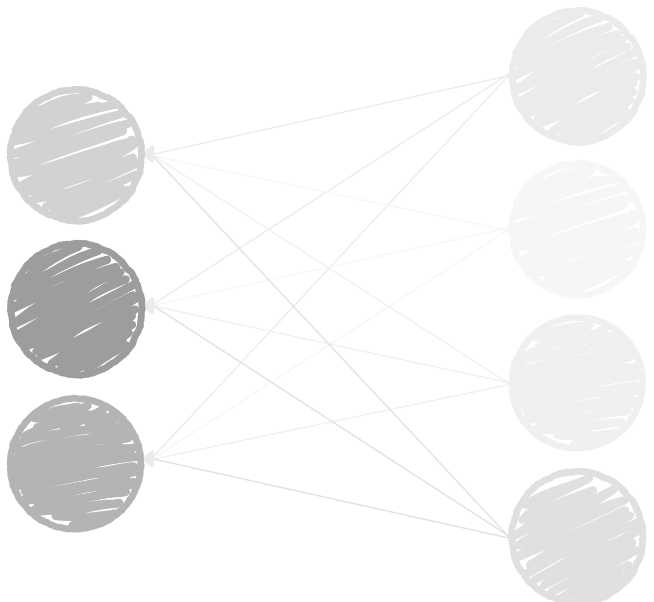


(B) Backward

Recreate Input



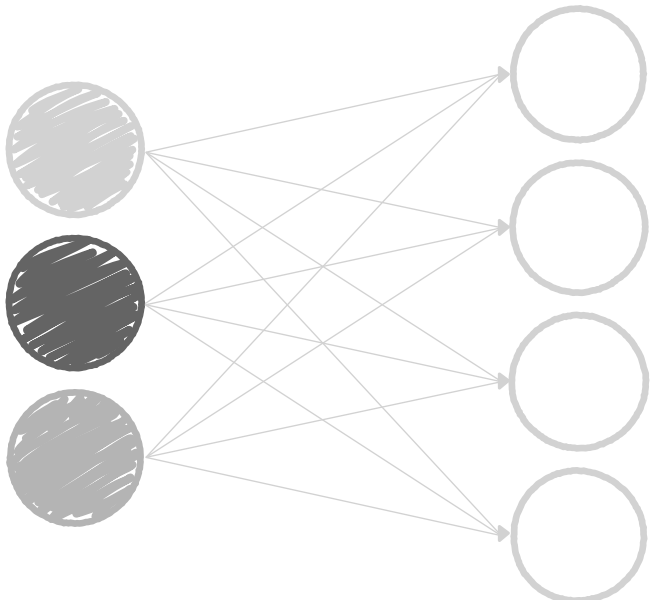
(A) Forward



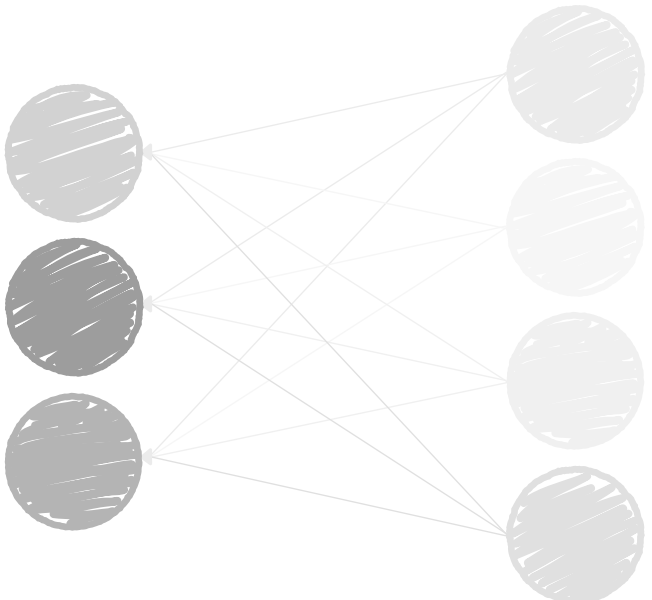
(B) Backward

Recreate Input

KL Divergence = Compare
Actual to Recreation



(A) Forward

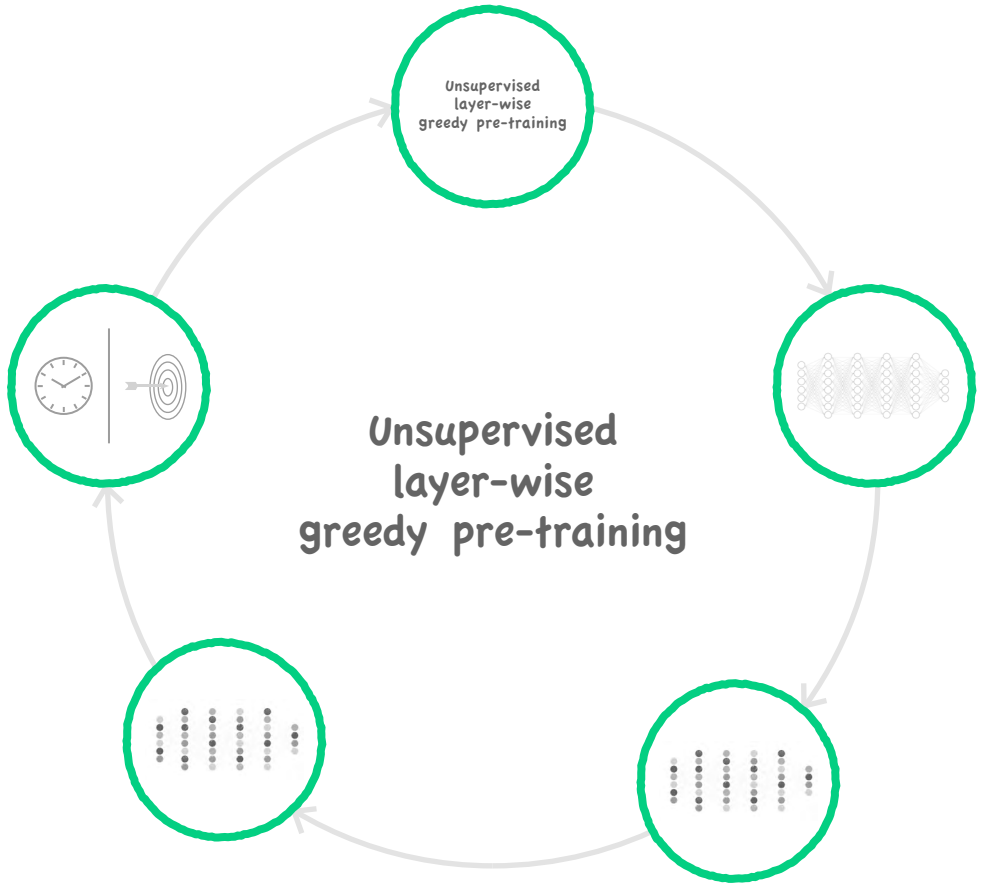


(B) Backward

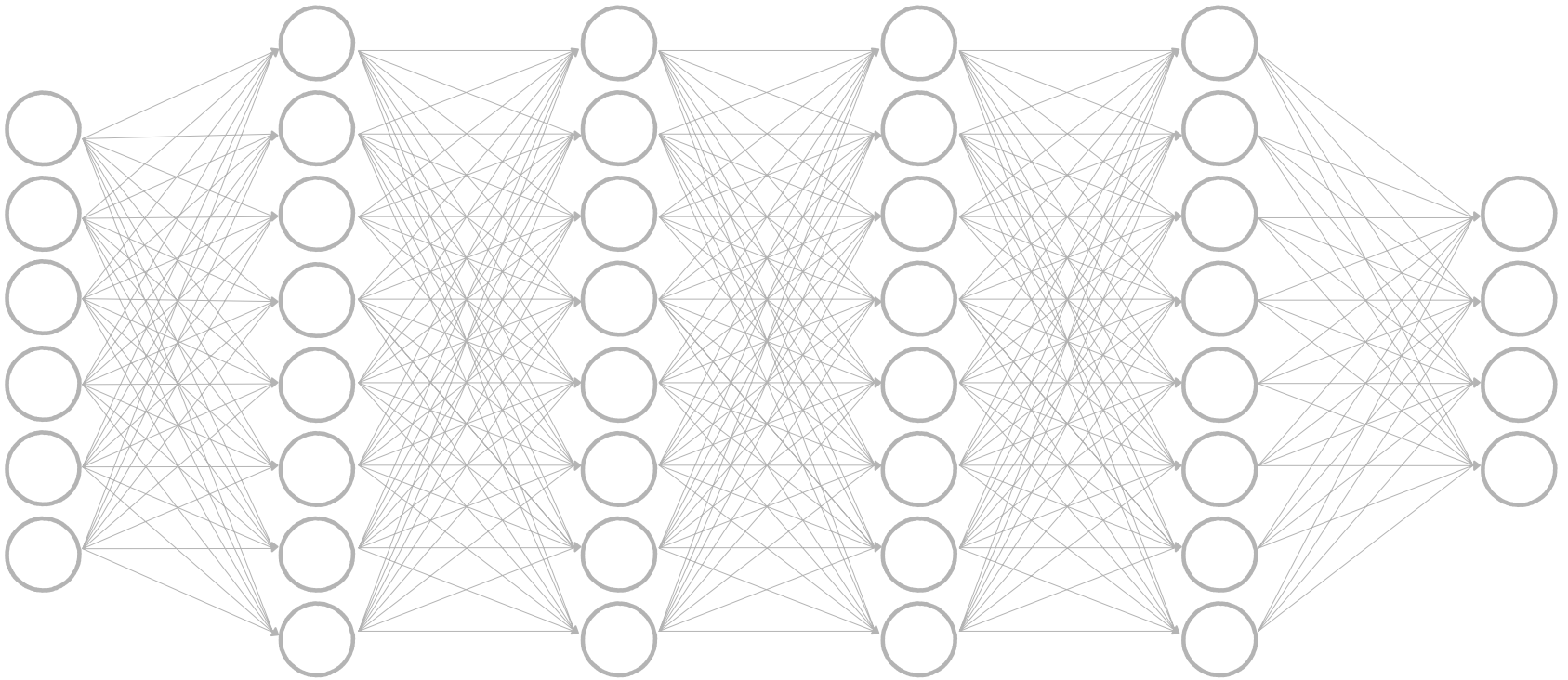
How Can We Use RBM to Initialize Weights?

- Apply the RBM idea on adjacent two layers as a pre-training step
- This will set weights
- Example : Deep Belief Network

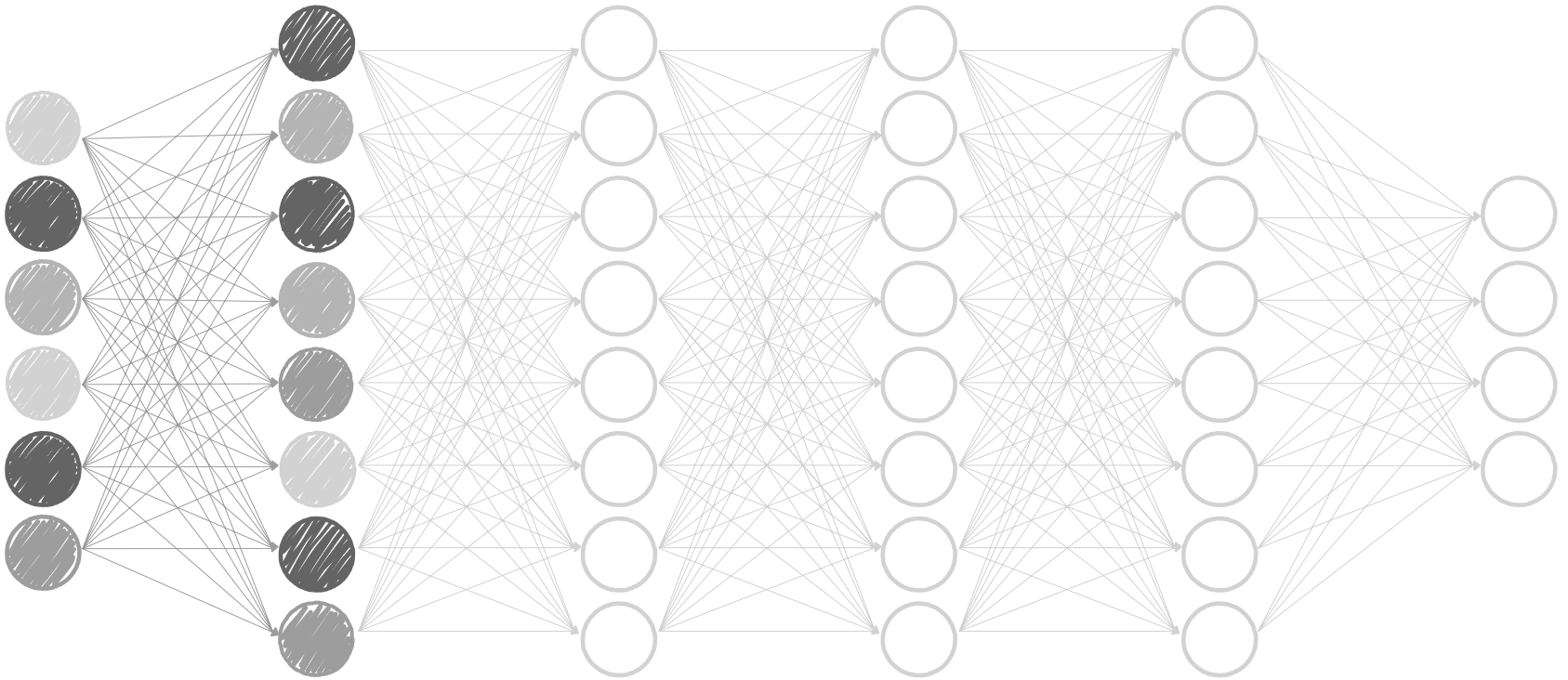
Deep Belief Network (DBN)



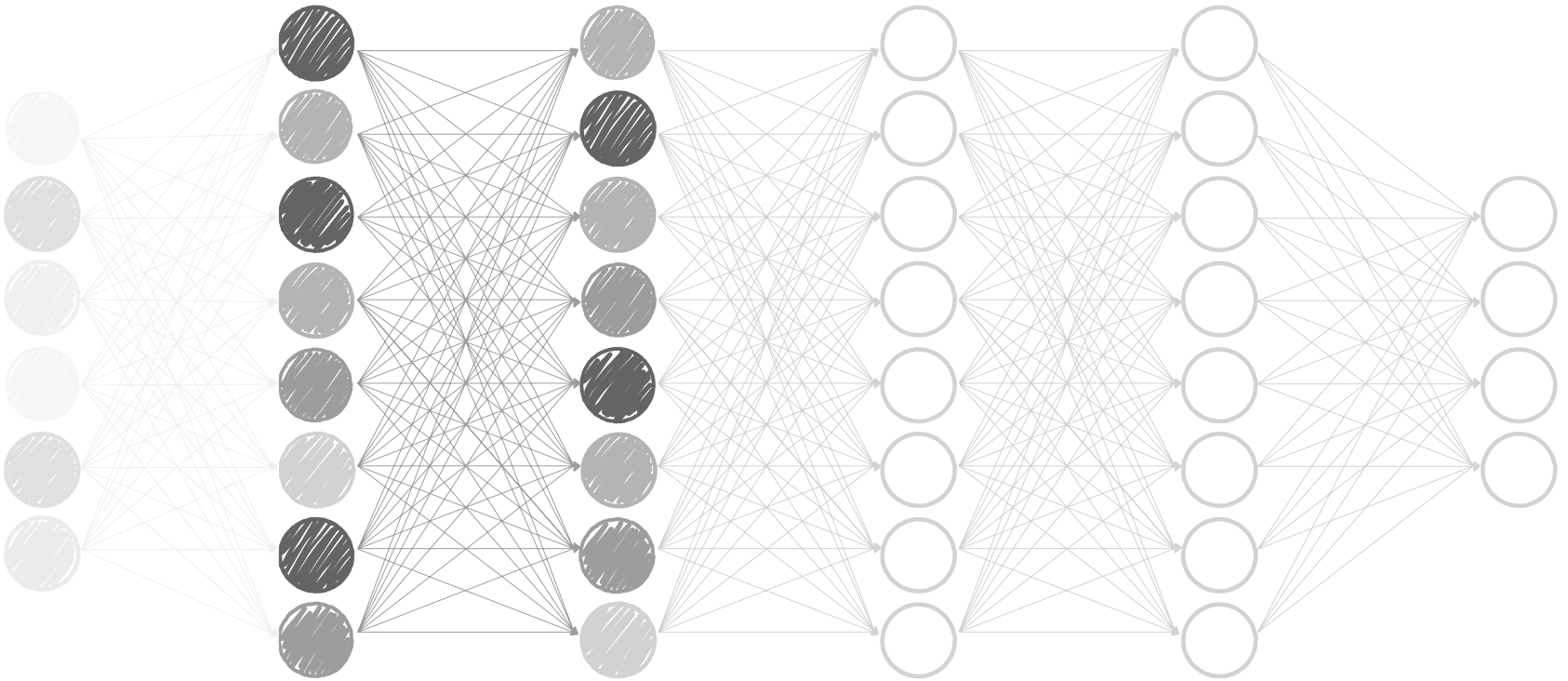
Pre-training



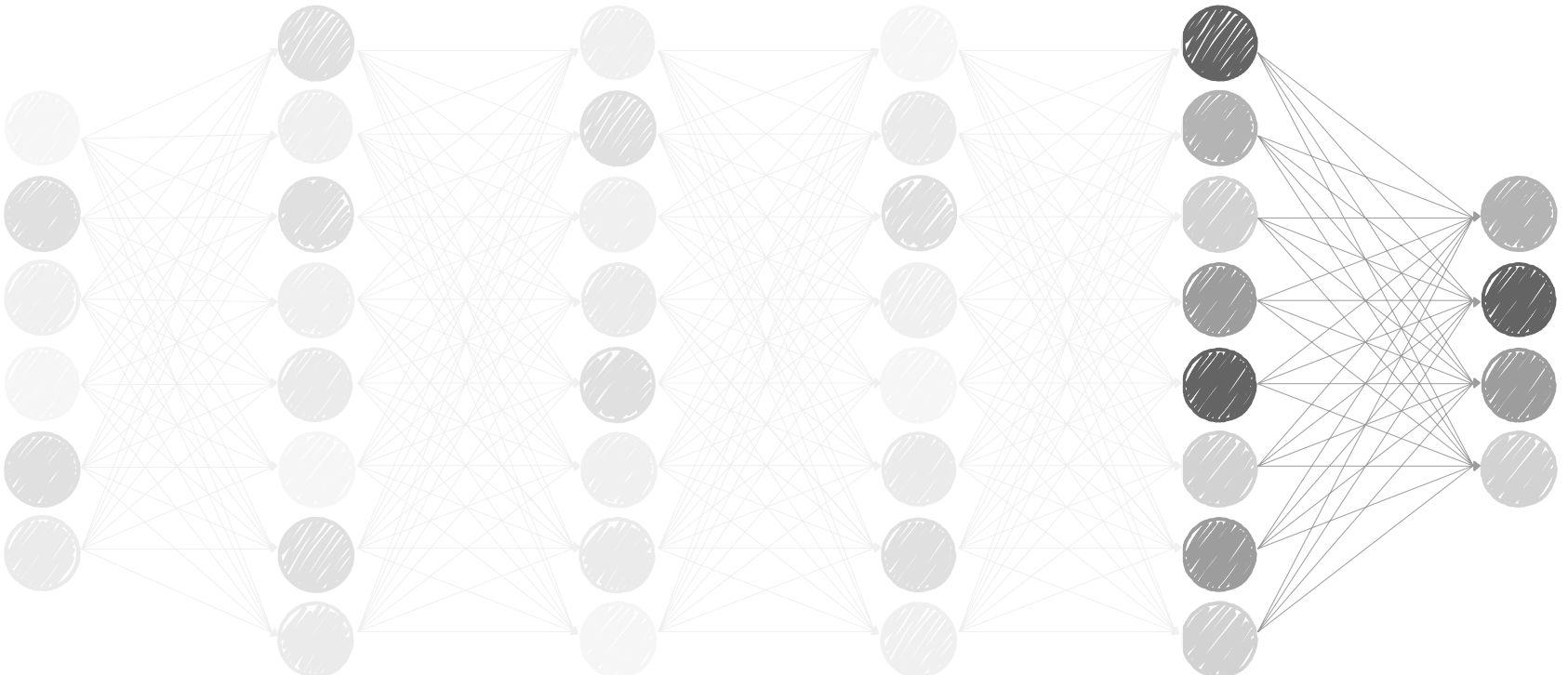
Pre-training



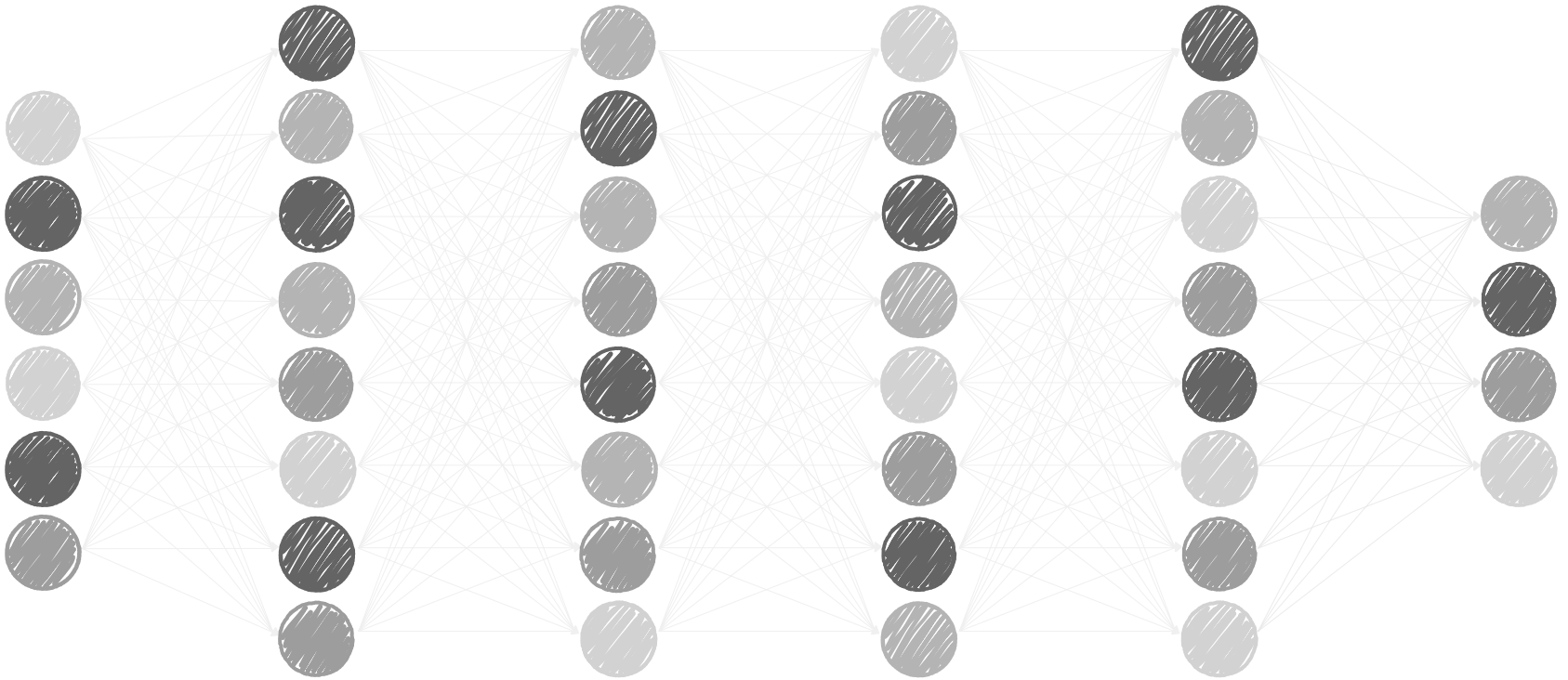
Pre-training



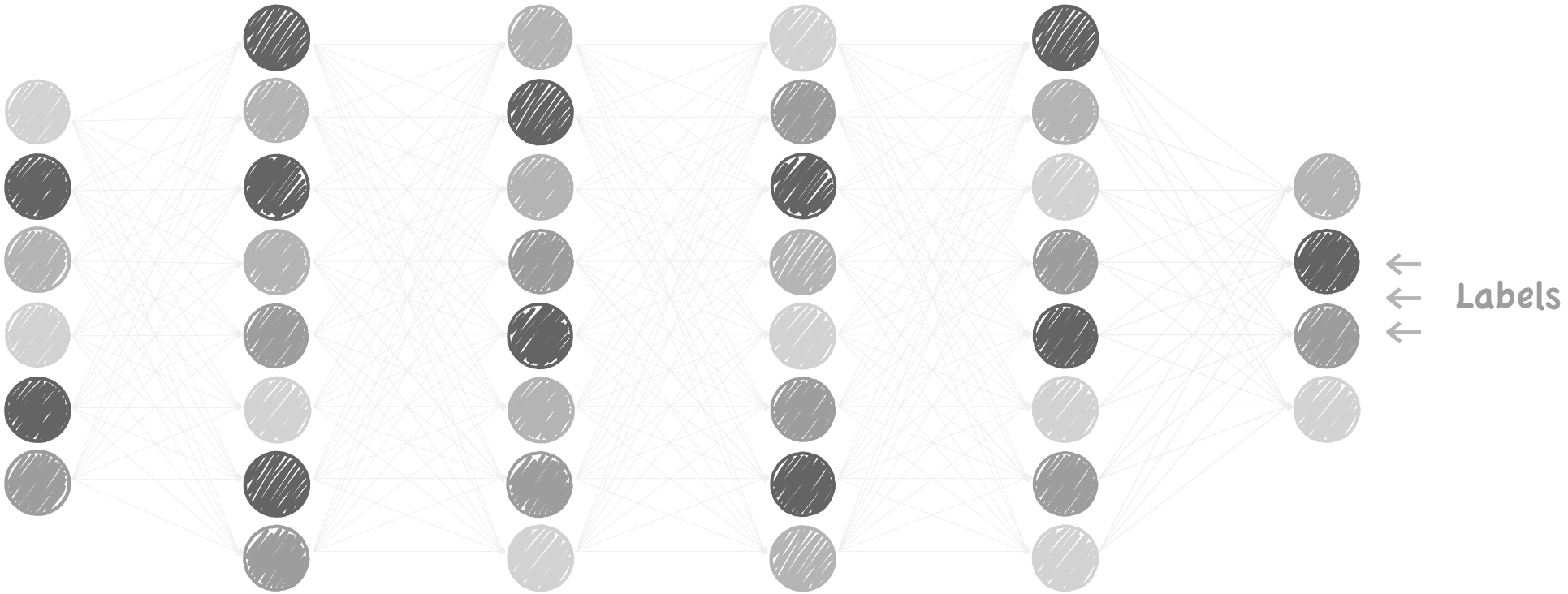
Pre-training



Pre-training



Fine Tuning



Good News

- No need to use complicated RBM for weight initializations

- Simple methods are OK

  01. Xavier initialization

  "Understanding the difficulty of training deep feedforward neural networks,"

  - X.Glorot and Y.Bengio, in International Conference on Artificial Intelligence and Statistics, 2010 -

  02. He's initialization

  "Delving Deep into Rectifiers:
  Surpassing Human-Level Performance on ImageNet Classification,"

  - K.He, X.Zhang, S.Ren, and J.Sun, 2015 -

Xavier Initialization

- Makes sure the weights are 'Just Right', not too Small, not too big

- Using number of input(fan_in) and output(fan_out)

```
# Xavier initialization
# Glorot et al. 2010
W = np.random.randn(fan_in, fan_out)/np.sqrt(fan_in)

# He et al. 2015
W = np.random.randn(fan_in, fan_out)/np.sqrt(fan_in/2)
```

Prettytensor Implementation

```
def xavier_init(n_inputs, n_outputs, uniform=True):
  """Set the parameter initialization using the method described.
  This method is designed to keep the scale of the gradients roughly the same
  in all layers.
  Xavier Glorot and Yoshua Bengio (2010):
           Understanding the difficulty of training deep feedforward neural
           networks. International conference on artificial intelligence and
           statistics.
  Args:
    n_inputs: The number of input nodes into each output.
    n_outputs: The number of output nodes for each input.
    uniform: If true use a uniform distribution, otherwise use a normal.
  Returns:
    An initializer.
  """
  if uniform:
    # 6 was used in the paper.
    init_range = math.sqrt(6.0 / (n_inputs + n_outputs))
    return tf.random_uniform_initializer(-init_range, init_range)
  else:
    # 3 gives us approximately the same limits as above since this repicks
    # values greater than 2 standard deviations from the mean.
    stddev = math.sqrt(3.0 / (n_inputs + n_outputs))
    return tf.truncated_normal_initializer(stddev=stddev)
```

http://stackoverflow.com/questions/33640581/how-to-do-xavier-initialization-on-tensorflow

Activation Functions and Initialization on CIFAR-10

Mishkin et al. 2015

Init method	maxout	ReLU	VLReLU	tanh	Sigmoid
LSUV	**93.94**	**92.11**	92.97	89.28	n/c
OrthoNorm	93.78	91.74	92.40	89.48	n/c
OrthoNorm-MSRA scaled	–	91.93	**93.09**	–	n/c
Xavier	91.75	90.63	92.27	**89.82**	n/c
MSRA	n/c†	90.91	92.43	89.54	n/c

Still an Active Area of Research

- We don't know how to initialize weight values, yet

- Many new algorithms
  - Batch normalization
  - Layer sequential uniform variance
  - ...

Geoffrey Hinton's Summary of Findings up to Today

- Our labeled datasets were thousands of times too small
- Our computers were millions of times too slow
- We initialized the weights in a stupid way
- We used the wrong type of non-linearity

http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning-part-4/

NEXT LECTURE

NN REGULARIZATION