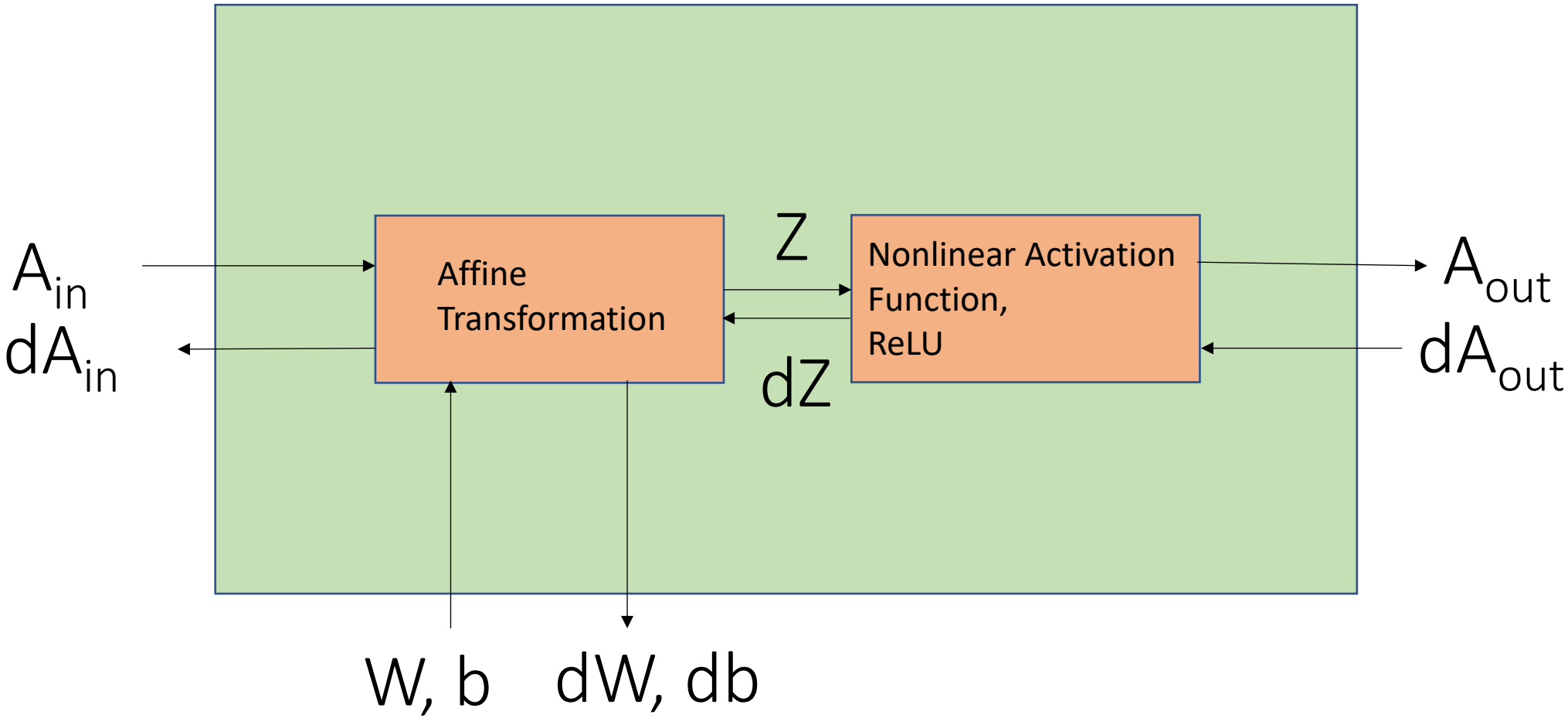
Deep Networks: Putting The Pieces Together

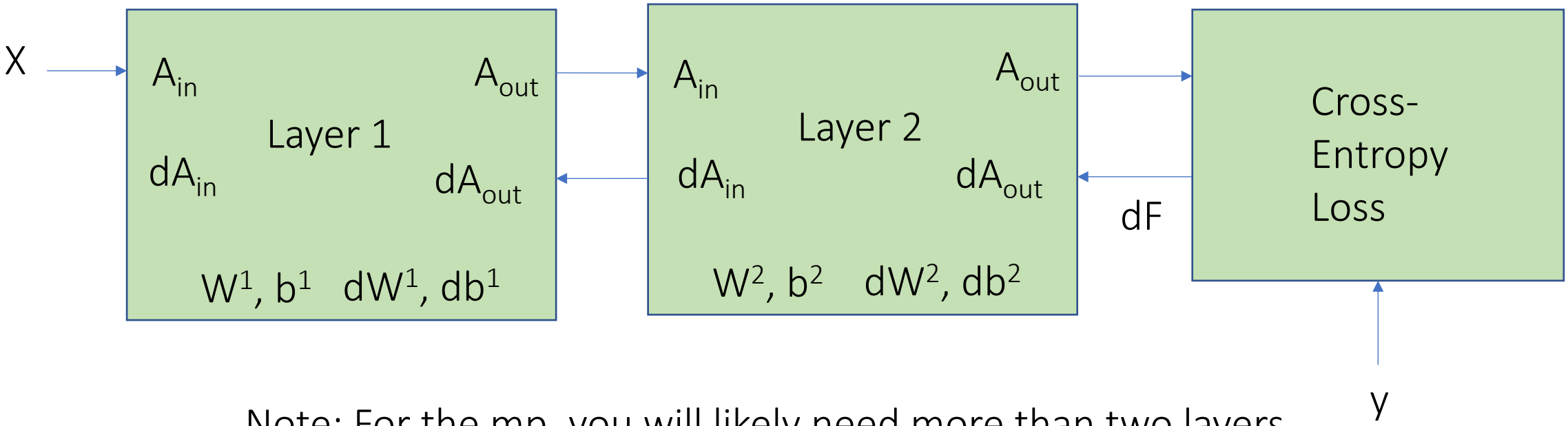
Neural Network Layer



Algorithm 1 Three Layer Network

procedure THREE-NETWORK($X, \{W^1, W^2, W^3\}, \{b^1, b^2, b^3\}, y$, test)
    Z^1, acache1 = AFFINE-FORWARD(X, W^1, b^1) ▷ acache = affine cache
    A^1, rcache1 = RELU-FORWARD(Z^1) ▷ rcache = relu cache
    Z^2, acache2 = AFFINE-FORWARD(A^1, W^2, b^2)
    A^2, rcache2 = RELU-FORWARD(Z^2, W^2, b^2)
    F, acache3 = AFFINE-FORWARD(A^2, W^3, b^3)
    if test == true **then**
        classifications = argmax over all classes in logits for each example
        return classifications
    loss, dF = CROSS-ENTROPY(F, y)
    dA^2, dW^3, db^3 = AFFINE-BACKWARD(dF, acache3)
    dZ^2 = RELU-BACKWARD(dA^2, rcache2)
    dA^1, dW^2, db^2 = AFFINE-BACKWARD(dZ^2, acache2)
    dZ^1 = RELU-BACKWARD(dA^1, rcache1)
    dX, dW^1, db^1 = AFFINE-BACKWARD(dZ^1, acache1)
    Use gradient descent to update parameters i.e. $W^1 = W^1$ - ηdW^1
    return loss

Neural Network Example and Computational Graph

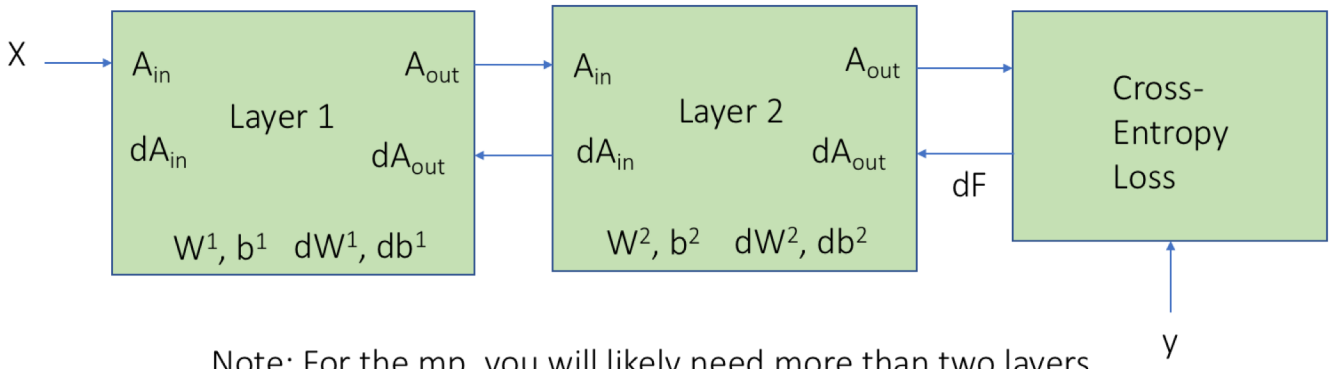


Note: For the mp, you will likely need more than two layers.

Tensorflow and Autdifferentitation

- Autodifferentiation: You only have to define the forward operation. The backwards operation will be automatically computed.

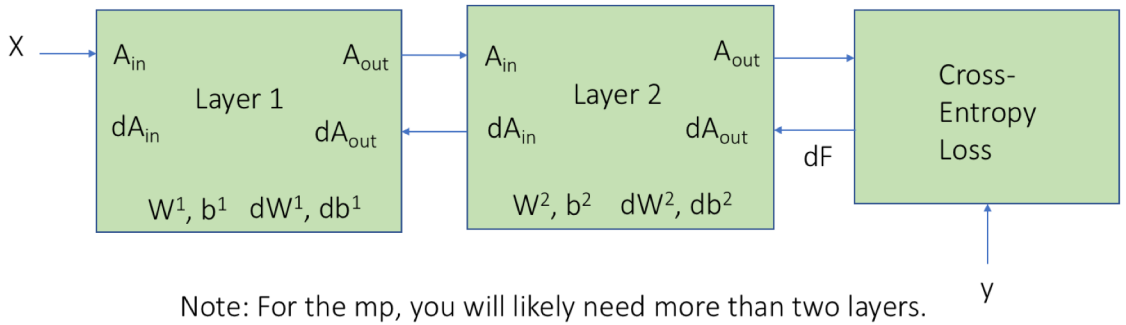
- **You build the computation graph first and then run the graph in a session.**



Note: For the mp, you will likely need more than two layers.

Tensorflow Example for Two Layer Net

```python
class TwoLayerNet(object):
    def __init__(self, units1, units2, lr):
        self.units1 = units1
        self.units2 = units2
        self.lr = lr

    def define_forward(self, x, y):
        # define graph but does not run graph!
        layer1 = tf.contrib.layers.fully_connected(x, self.units1,
                                                   activation_fn=tf.nn.relu)
        output = tf.contrib.layers.fully_connected(layer1, self.units2,
                                                   activation_fn=None)
        loss = tf.nn.softmax_cross_entropy_with_logits(logits=output, labels=y)
        train_op = tf.train.GradientDescentOptimizer(self.lr).minimize(loss)
        return output, train_op
```



Note: For the mp, you will likely need more than two layers.

Tensorflow Example Continued

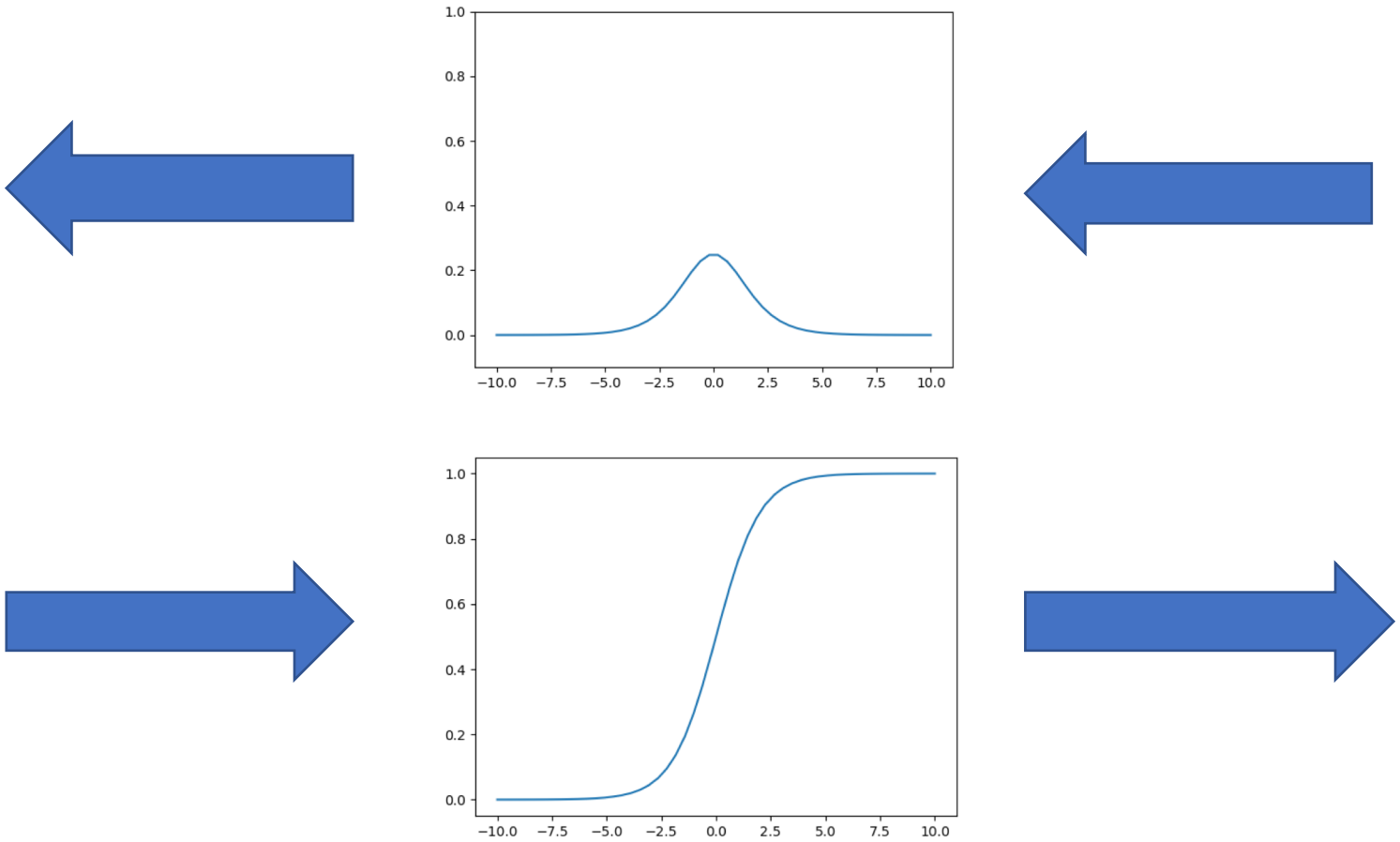
```python
def main():
    # load dataset
    mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)

    # Create model
    net = TwoLayerNet(128, 10, 0.01) # hidden layer has 100 units, last layer has 10
    x = tf.placeholder(tf.float32, [None, 784]) # placeholder for our data
    y = tf.placeholder(tf.float32, [None, 10])
    output, train_op = net.define_forward(x, y)

    # Run in session
    with tf.Session() as sess:
        # Train model for 1000 iterations
        sess.run(tf.global_variables_initializer())
        for _ in range(1000):
            bx, by = mnist.train.next_batch(100)
            scores, _ = sess.run([output, train_op], feed_dict={x: bx, y: by})
            accuracy = calculate_accuracy(by, scores)
            print('Training Accuracy: ', accuracy)

        # Evaluate model
        scores = sess.run(output, {x: mnist.test.images})
        accuracy = calculate_accuracy(mnist.test.labels, scores)
        print('Testing Accuracy: ', accuracy)
```

Tensorflow Ops

- Placeholders
- Variables
- Math Operations

Why it is important to know backpropagation?

- Why do we need to know backpropagation? It helps us avoid bugs!



Terminology of Reinforcement Learning

- Policy function: function that maps a state to an action.
- Q-Function: expected future reward for states and actions.
- Value Function: expected future reward for states.
- Deep learning can learn and approximate functions.

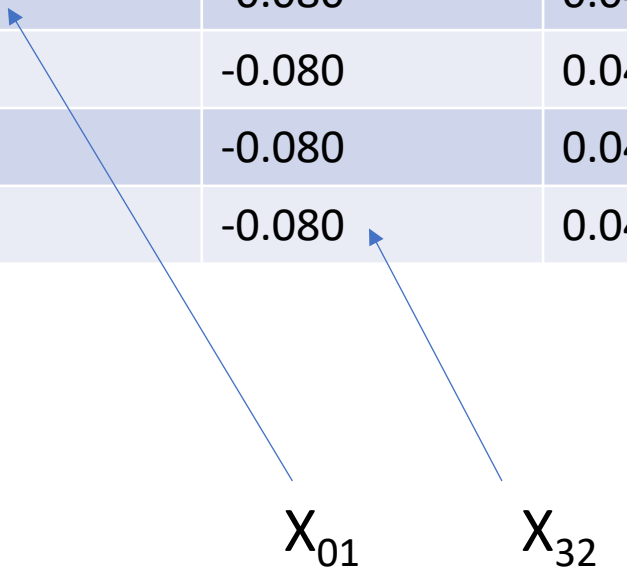
Dataset for MP4

Batch Data X

	Ball-X	Ball-Y	Velocity-X	Velocity-Y	Paddle-Y
X_0	0.296	0.265	-0.080	0.049	0.120
X_1	0.216	0.315	-0.080	0.049	0.080
X_2	0.136	0.364	-0.080	0.049	0.120
X_3	0.056	0.413	-0.080	0.049	0.120

Batch Targets y

Action	
0	y_0
2	y_1
1	y_2
1	y_3

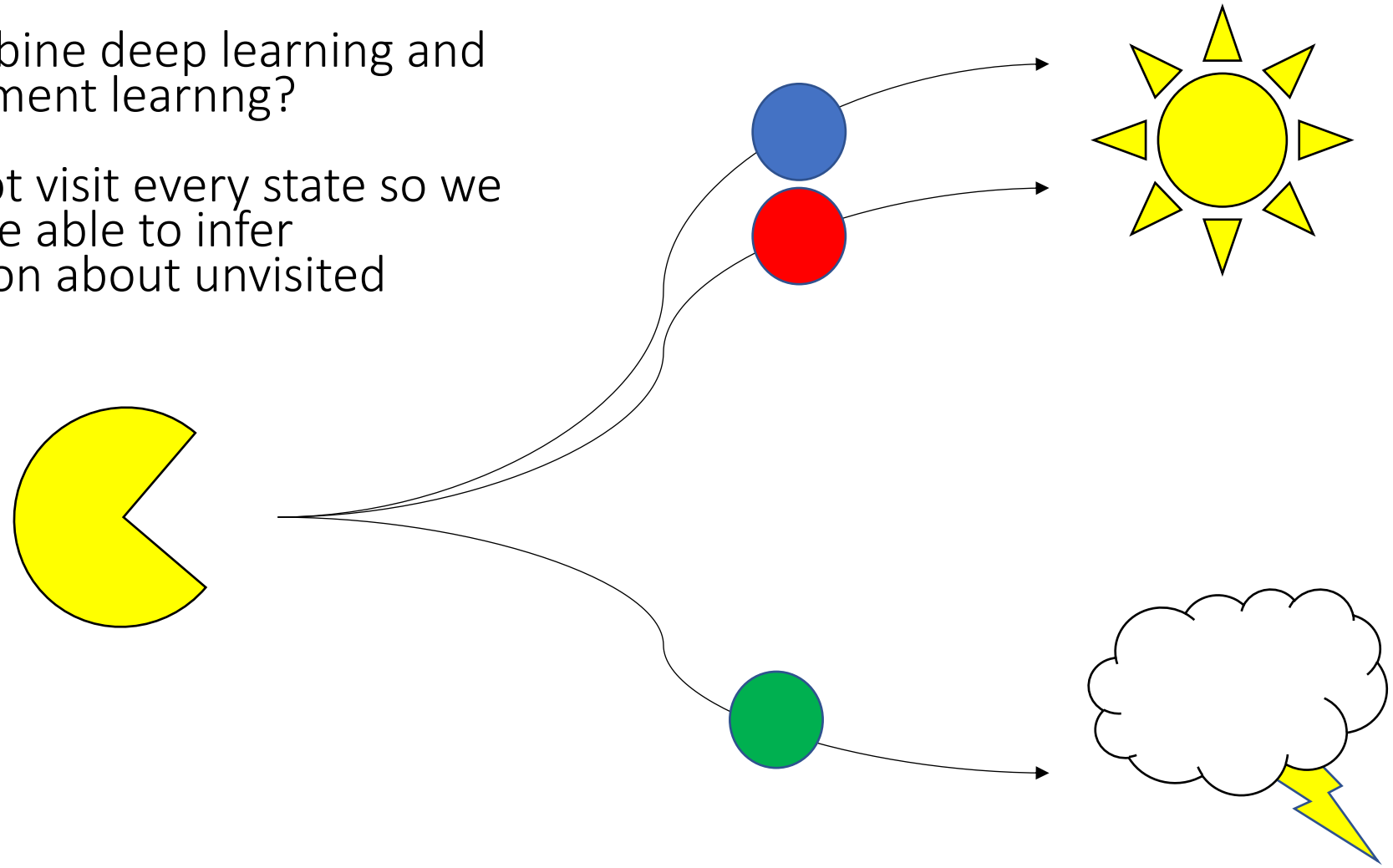


Behavioral Cloning / Imitation Learning
Approximate the policy function of an expert.

What are the advantages of deep learning?

Why combine deep learning and reinforcement learnng?

We cannot visit every state so we need to be able to infer information about unvisited states.



Up next: Deep Q-Learning