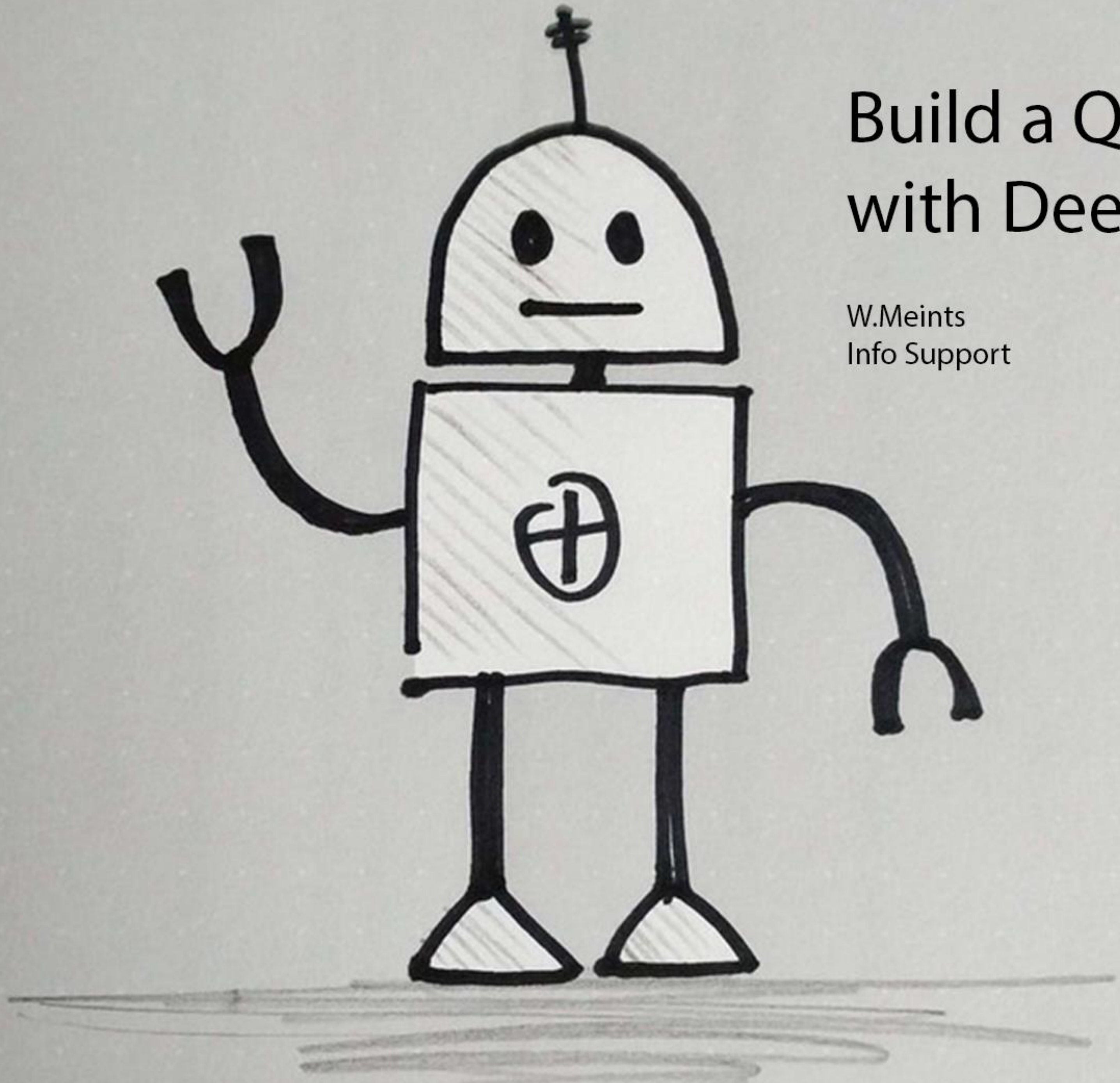


Build a Q&A bot with DeepLearning4J

W.Meints
Info Support

Please

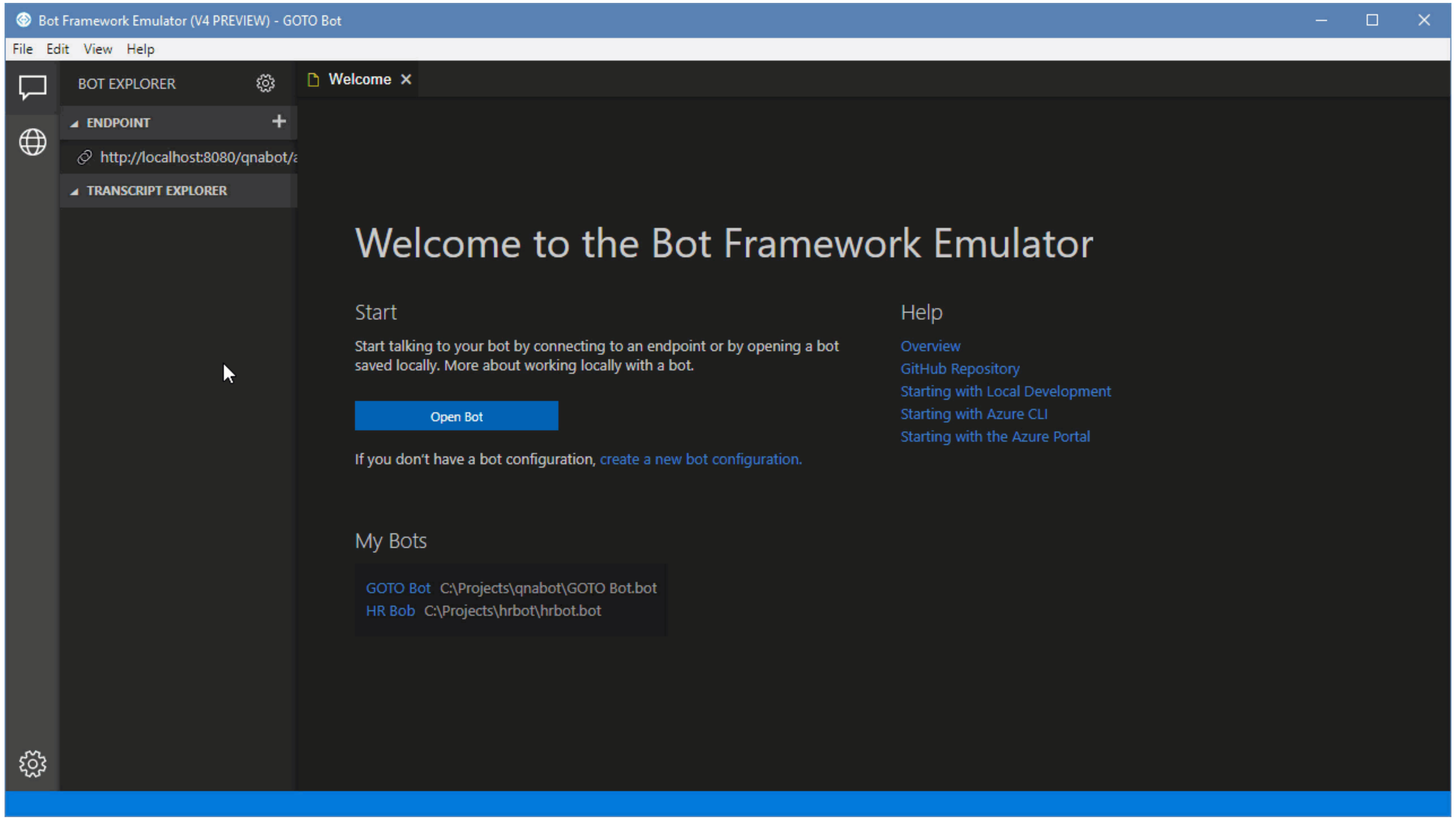
Ask questions through the app

Rate Session

Thank you!

Agenda

- Overview of deep learning
- Building a FAQ model with DeepLearning4J
- Integrating with a chatbot application


Bot Framework Emulator (V4 PREVIEW) - GOTO Bot
File
Edit
View
Help
BOT EXPLORER
ENDPOINT
http://localhost:8080/qnabot/a
TRANSCRIPT EXPLORER
Welcome

Welcome to the Bot Framework Emulator

Start

Start talking to your bot by connecting to an endpoint or by opening a bot saved locally. More about working locally with a bot.

Open Bot

If you don't have a bot configuration, create a new bot configuration.

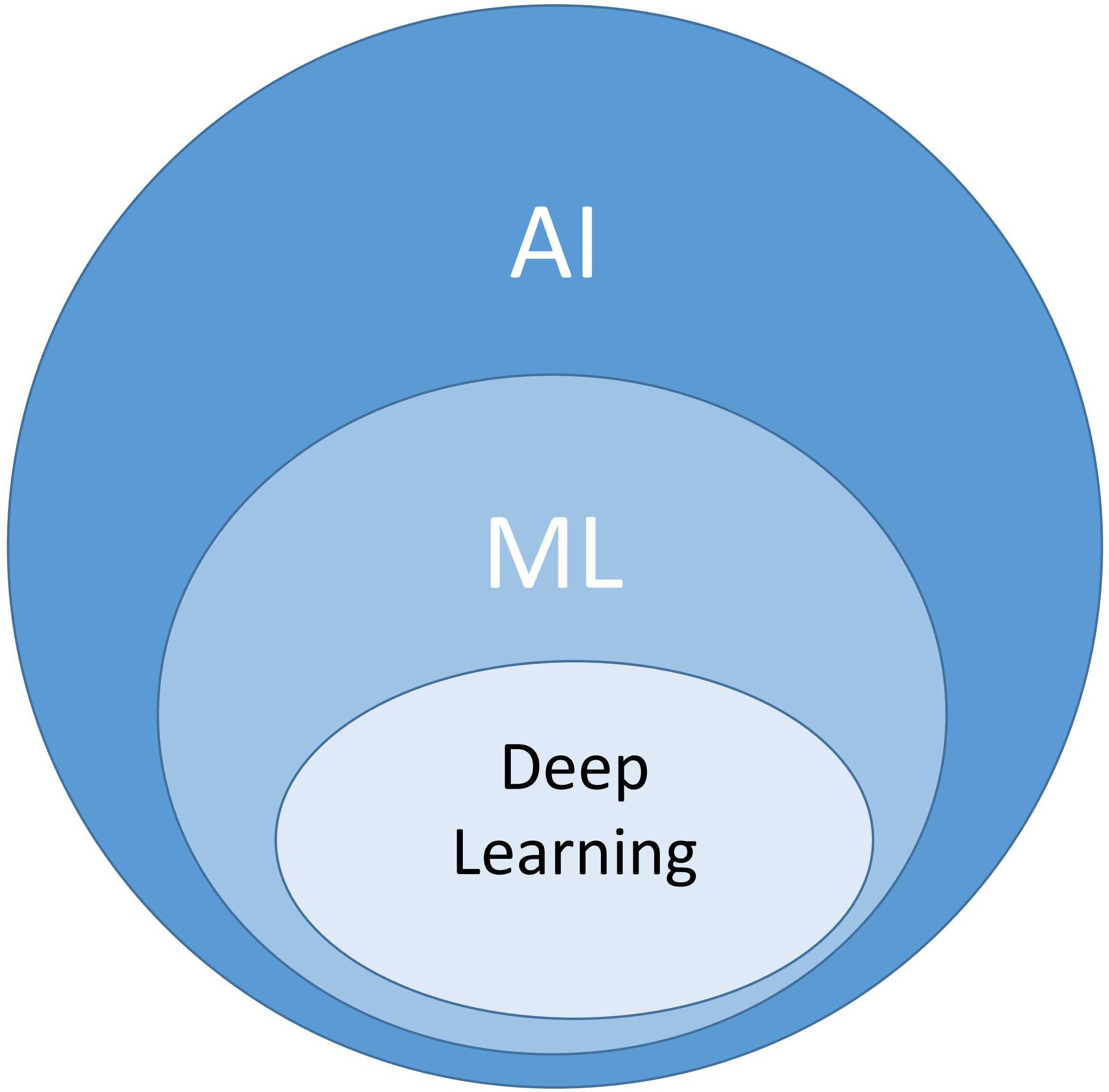
My Bots

GOTO Bot C:\Projects\qnabot\GOTO Bot.bot
HR Bob C:\Projects\hrbot\hrbot.bot

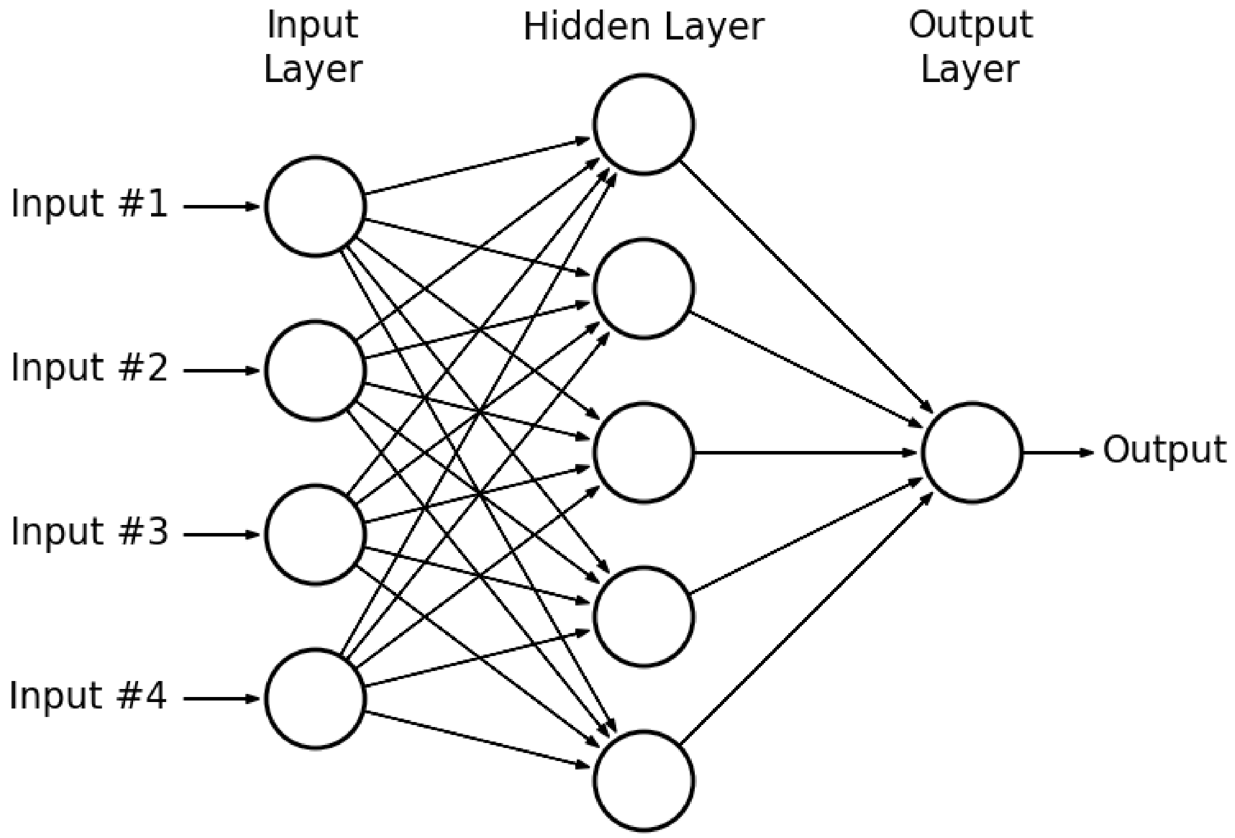
Help

Overview
GitHub Repository
Starting with Local Development
Starting with Azure CLI
Starting with the Azure Portal

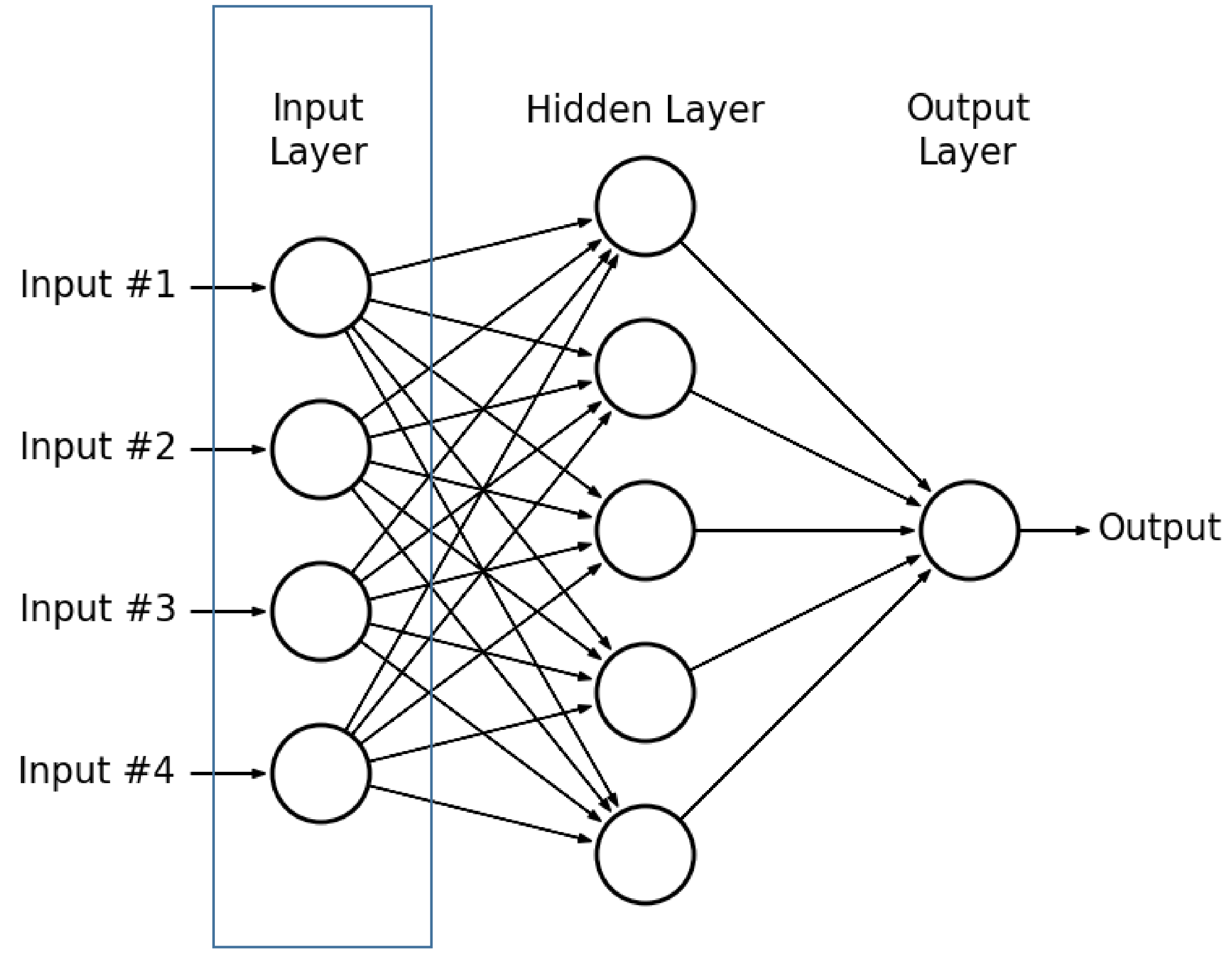
Overview of deep learning


AI
ML
Deep Learning

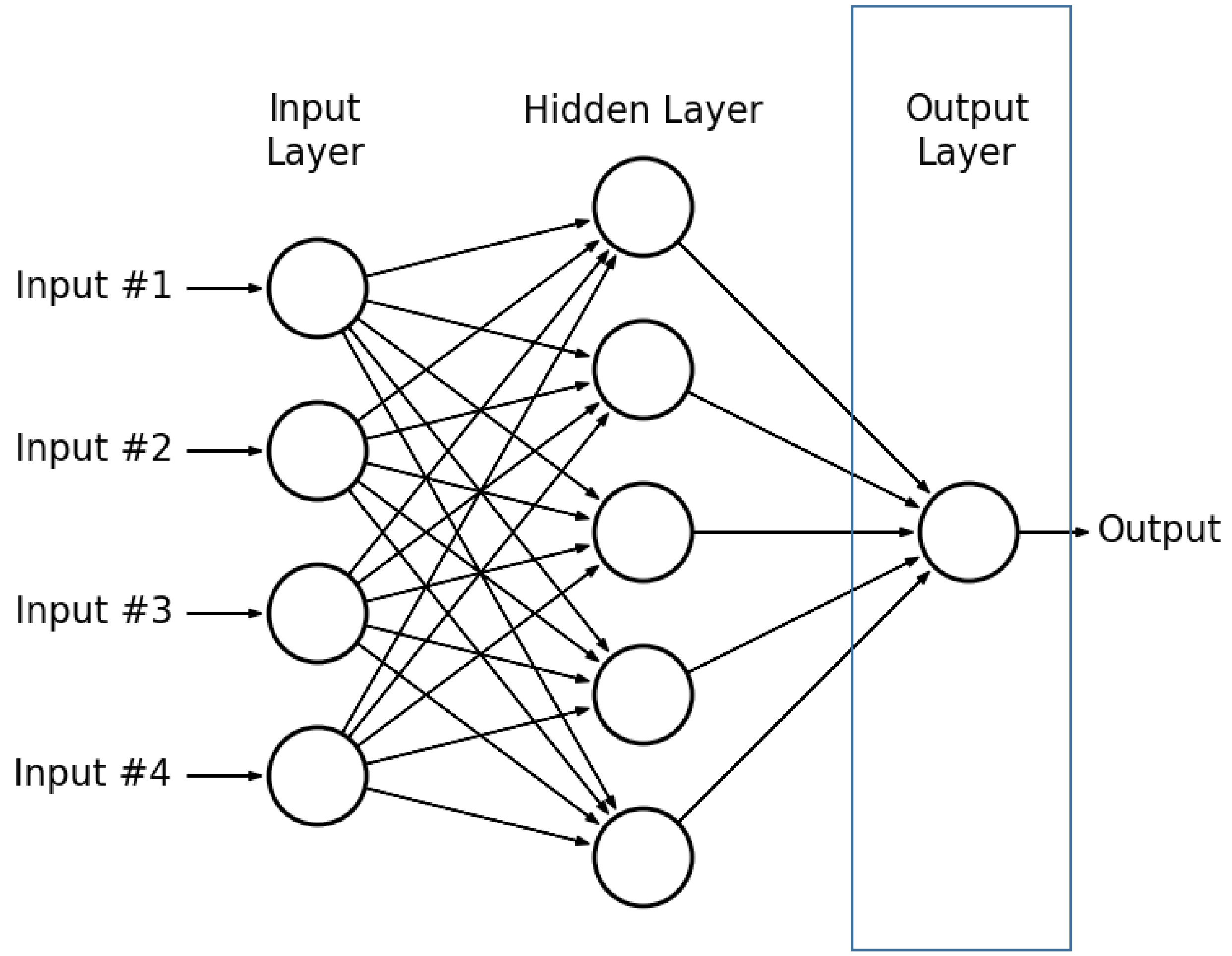
Neural network architecture



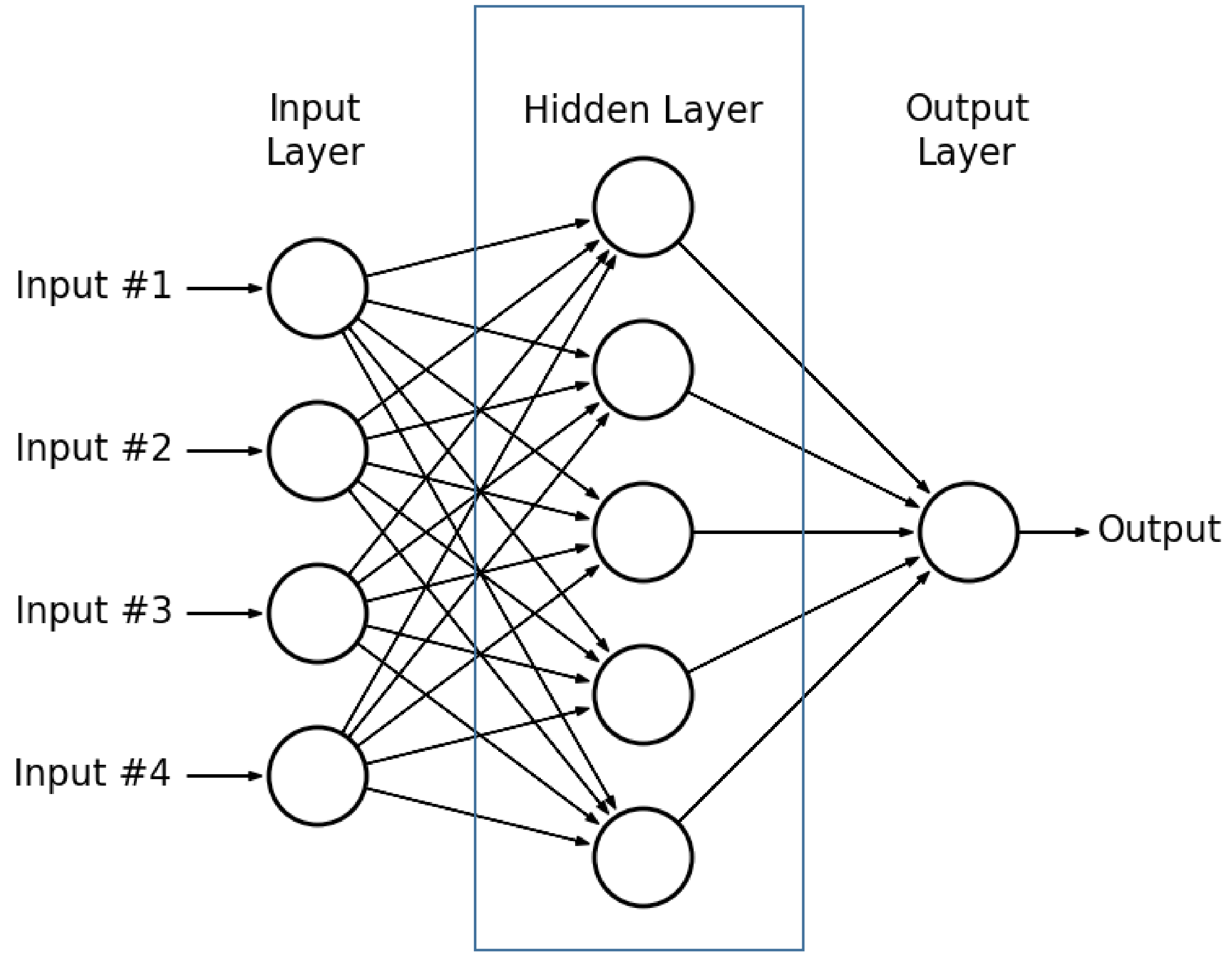
Neural network architecture



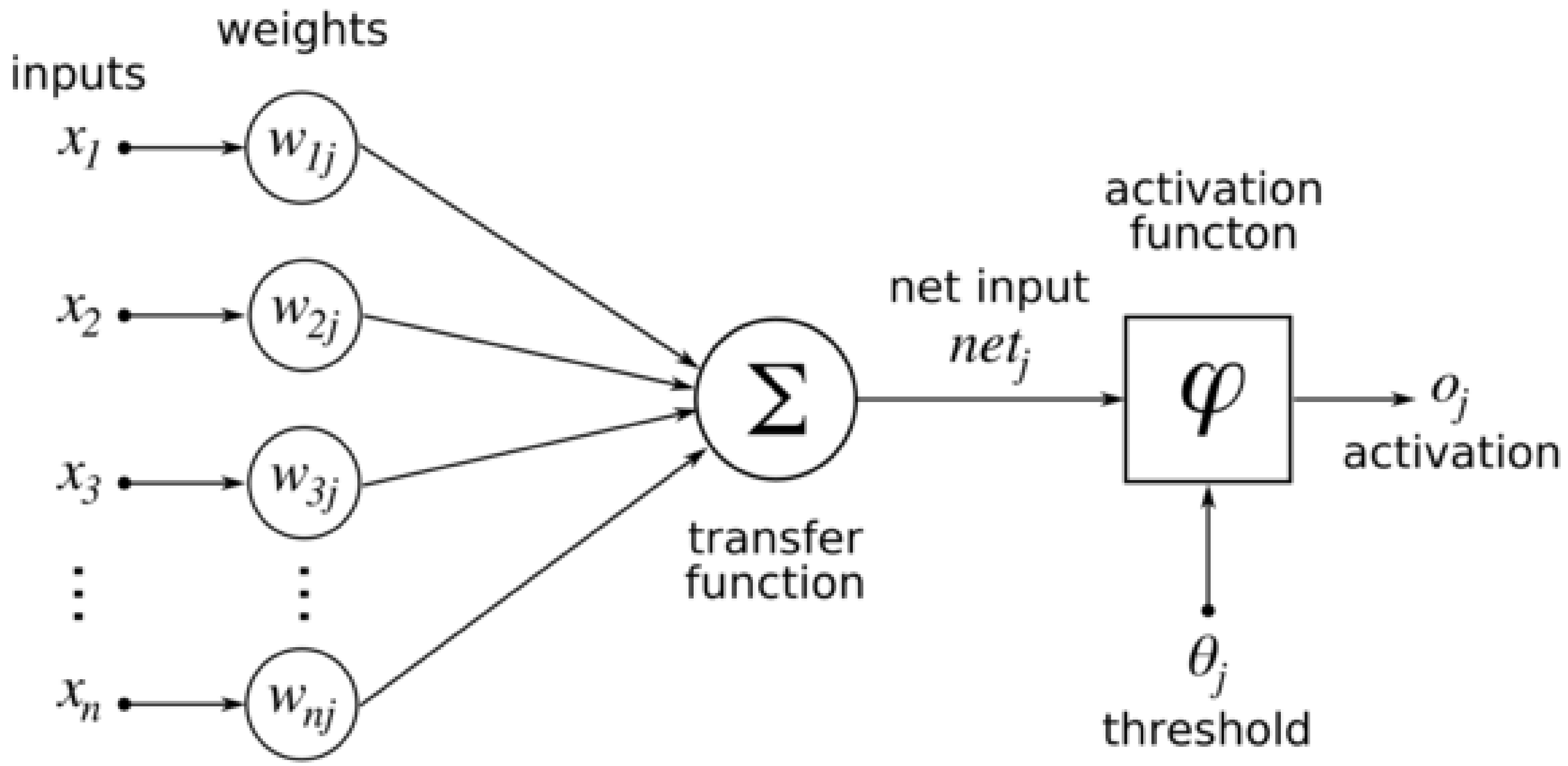
Neural network architecture



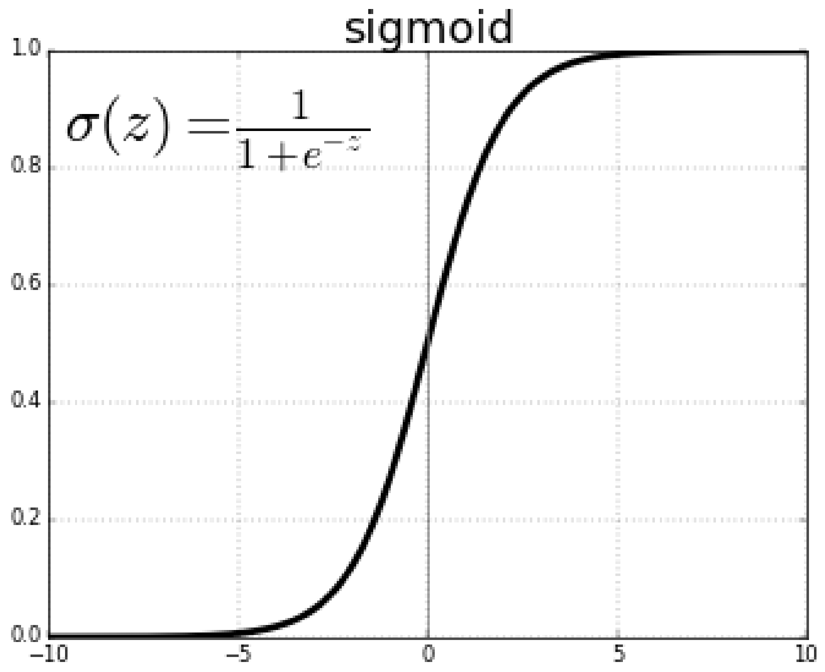
Neural network architecture

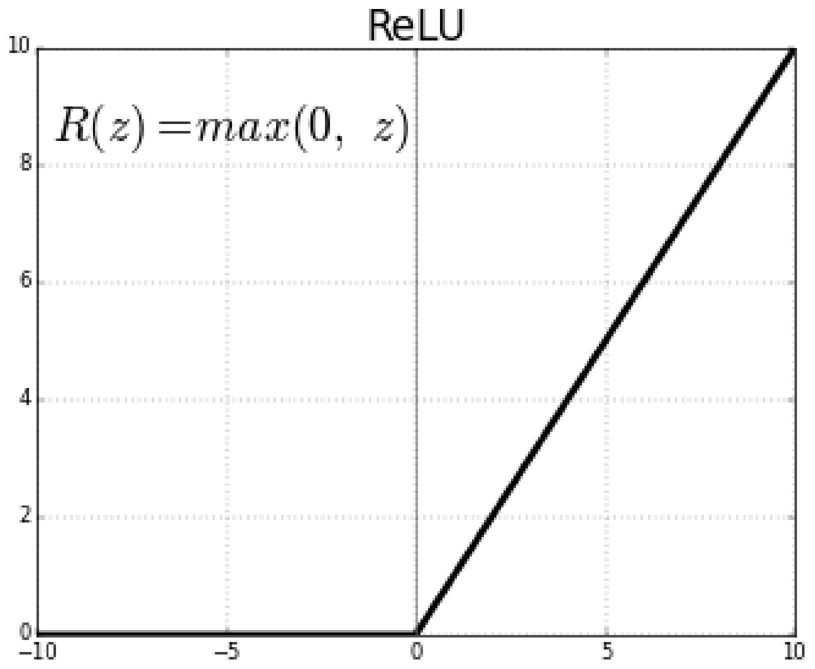


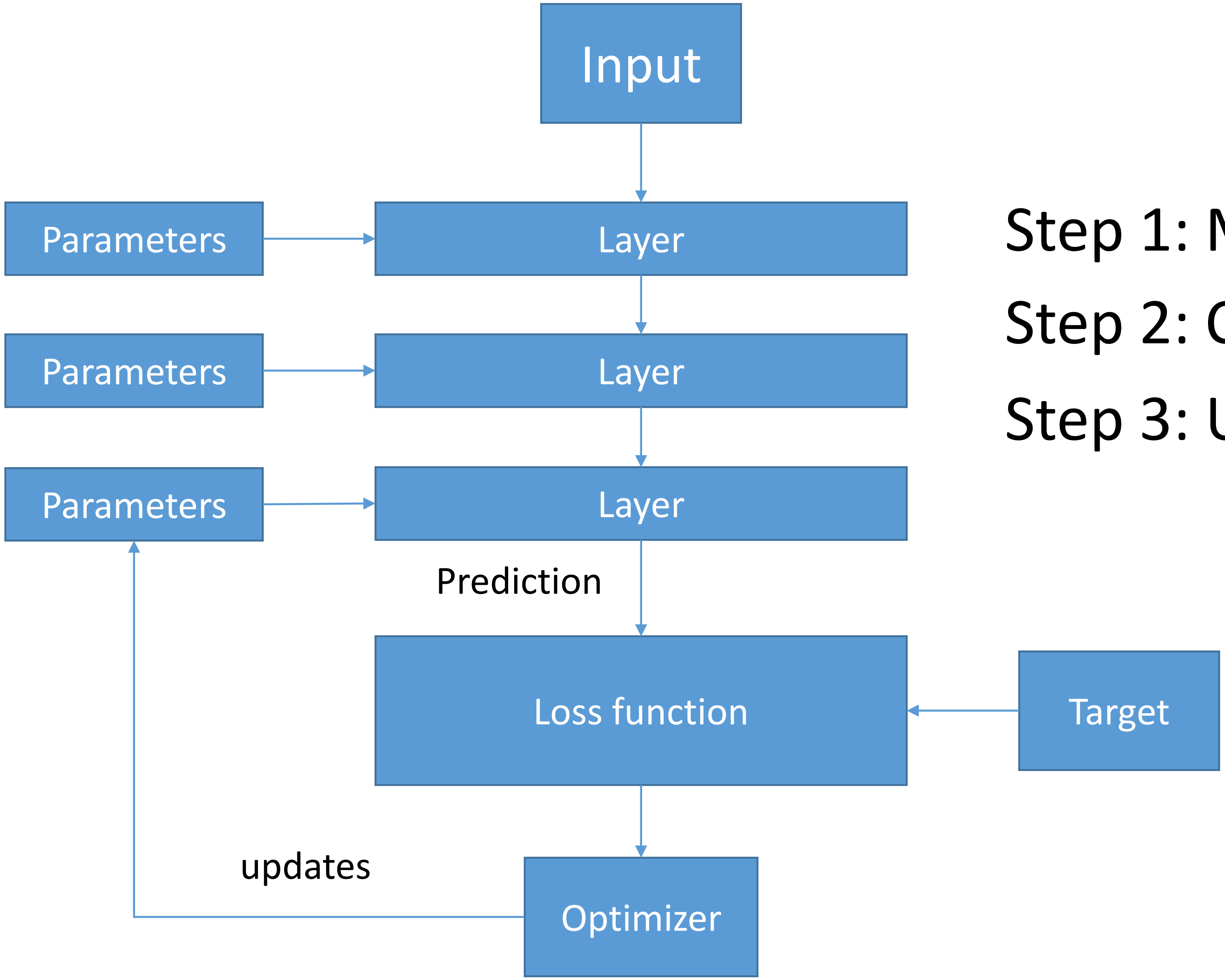
What happens inside a neuron



The role of activation functions







Step 1: Make a prediction

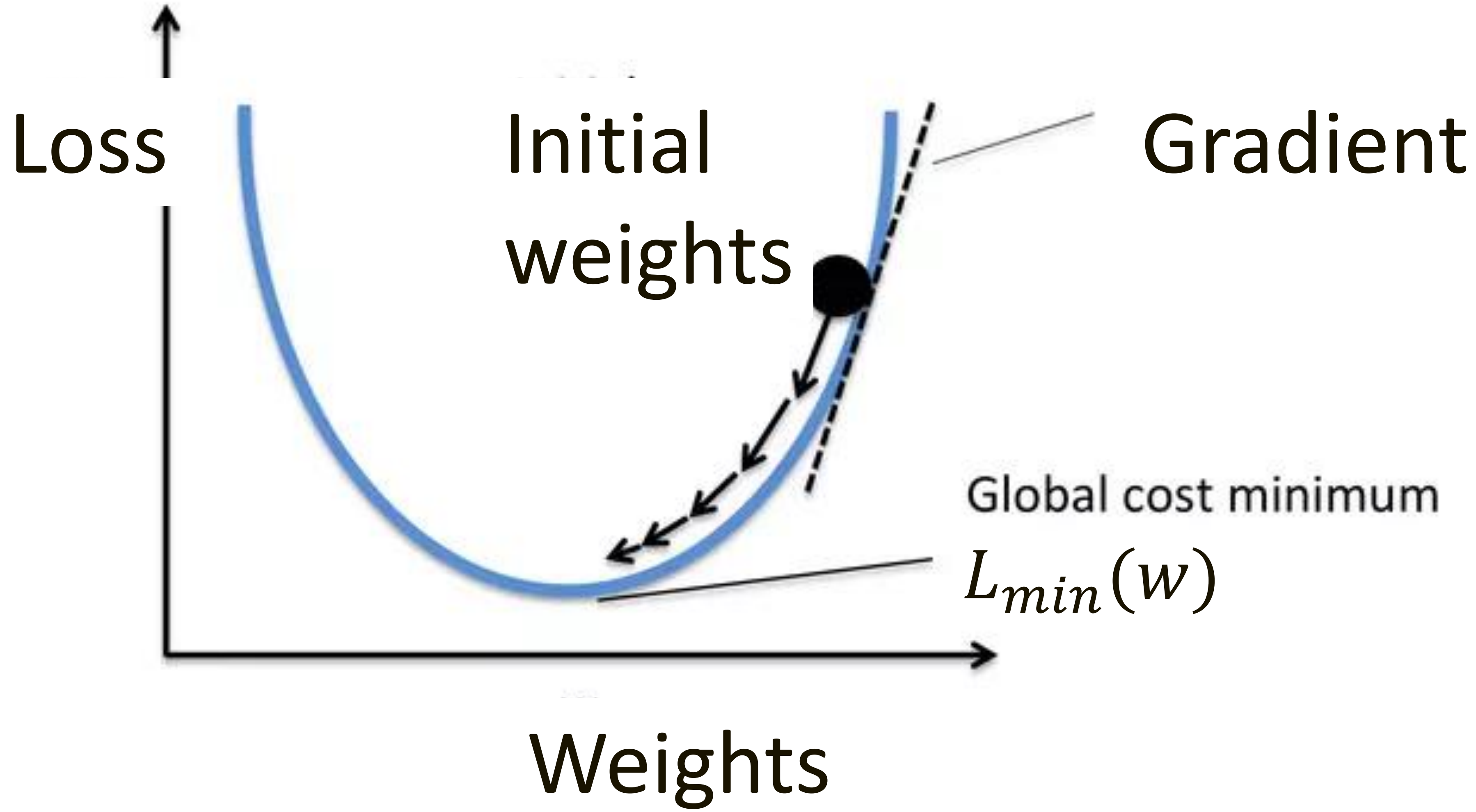
Step 2: Calculate loss

Step 3: Update weights

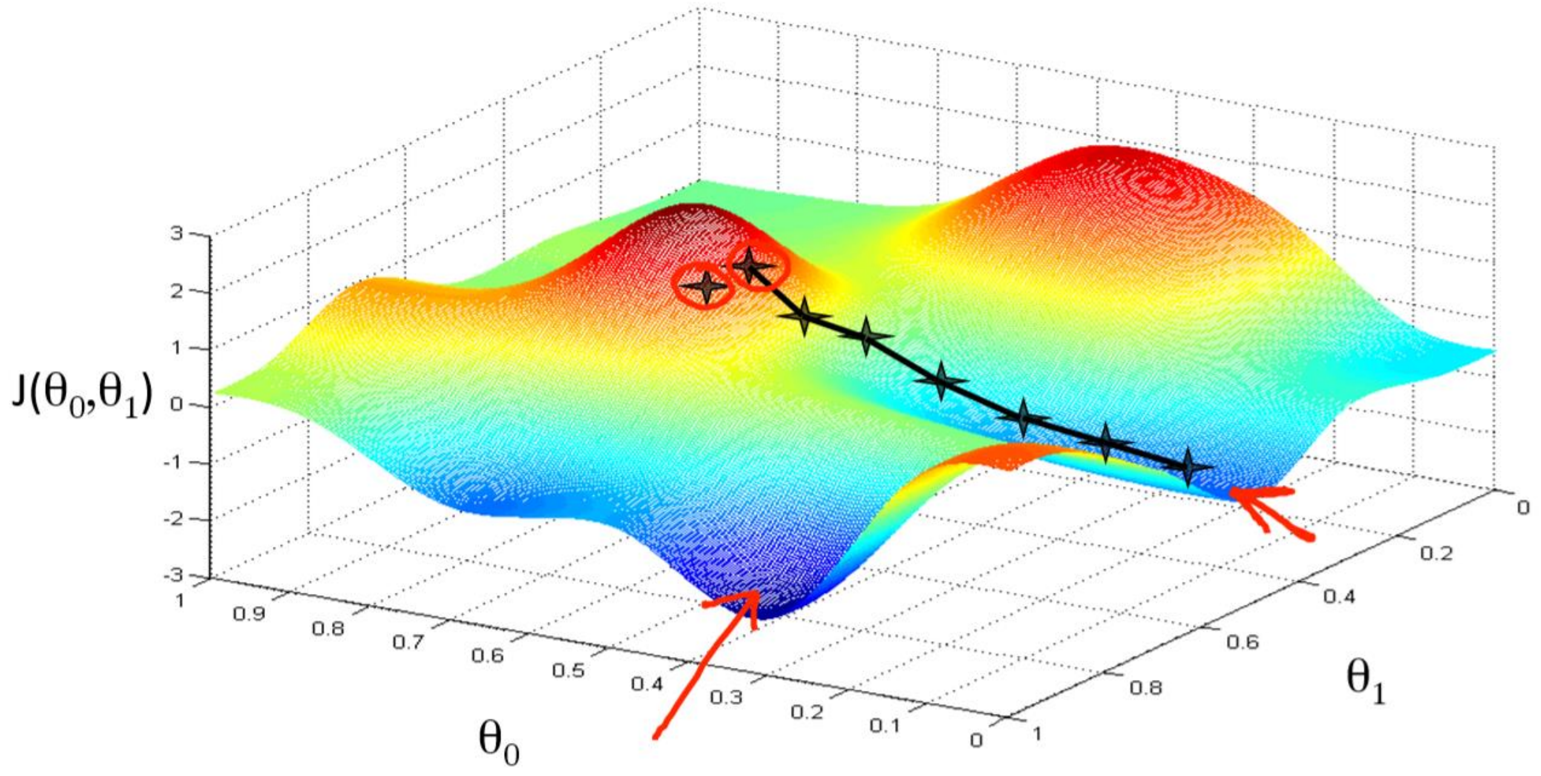
Loss is calculated using a loss function

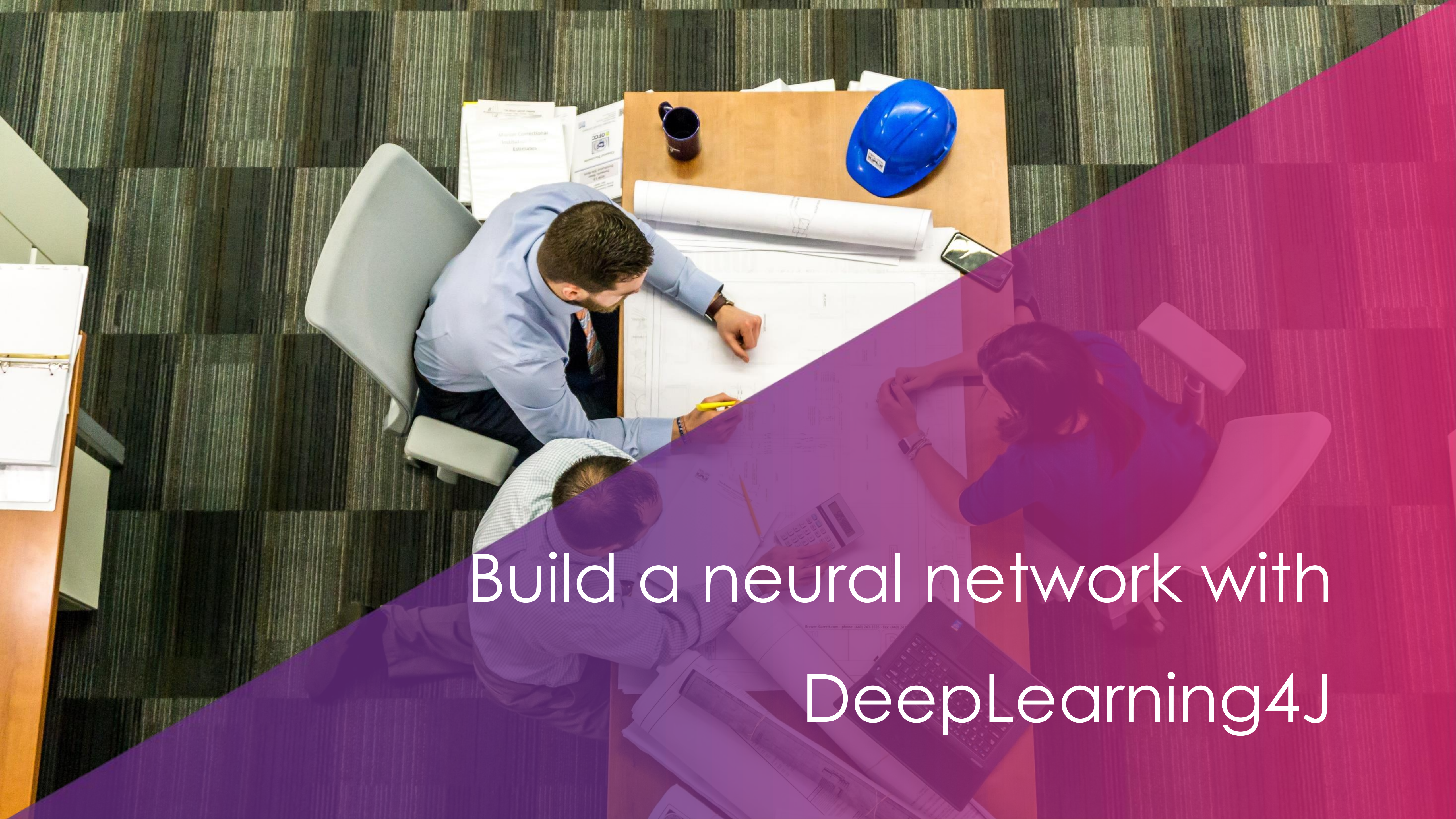
$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = -\frac{1}{N}\sum_{i}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)]$$

»


Loss
Initial weights
Gradient
Global cost minimum
$L_{min}(w)$
Weights

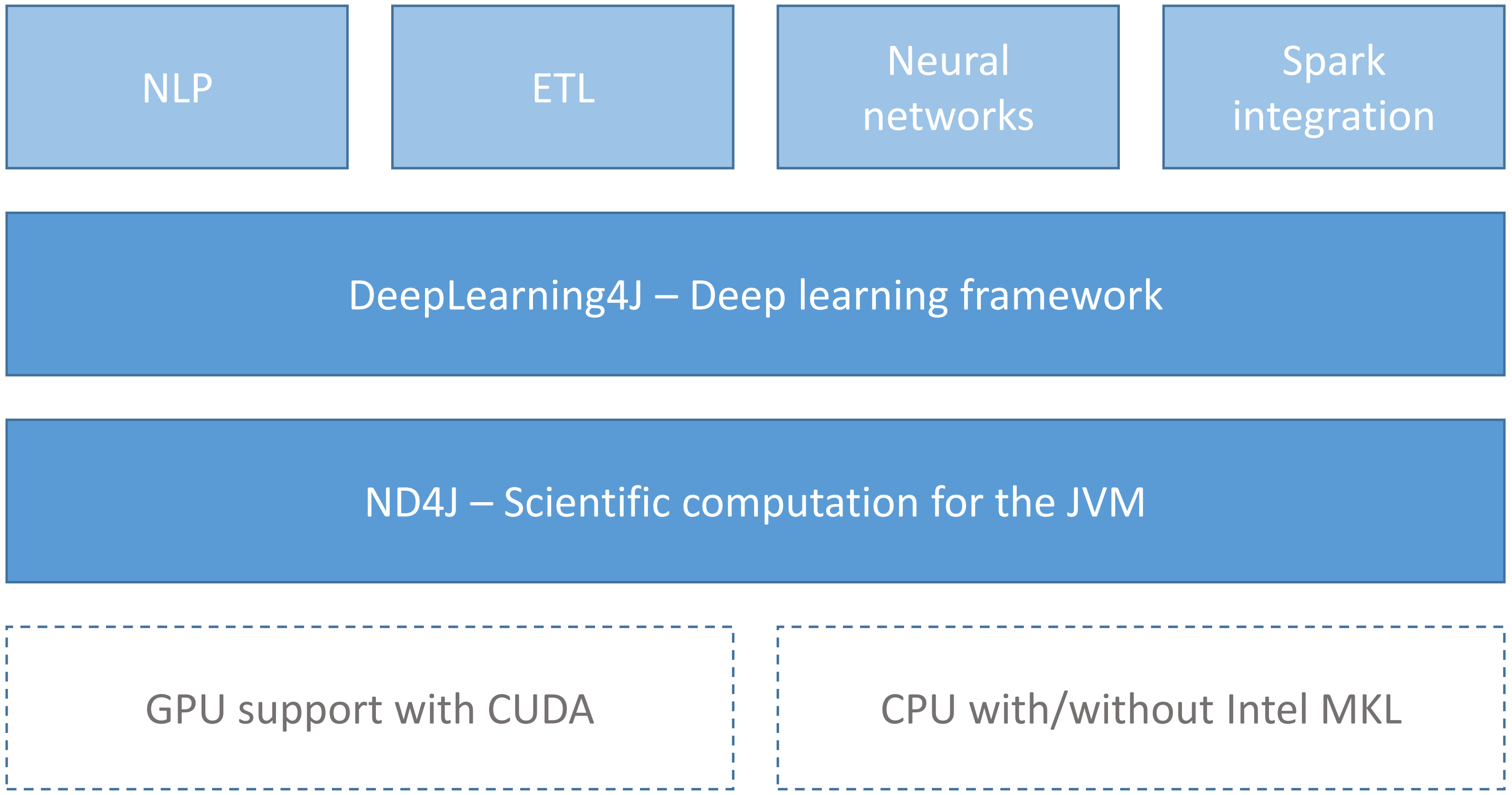
Gradient descent is not perfect!





Build a neural network with DeepLearning4J

DL4J

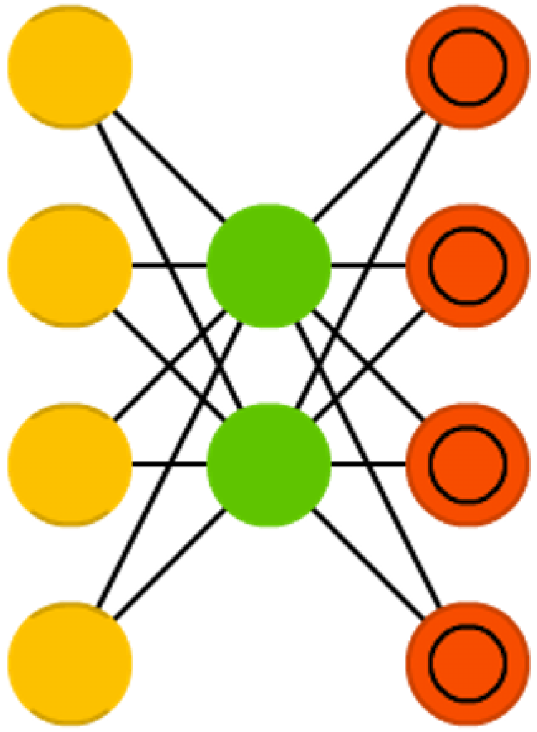

NLP
ETL
Neural networks
Spark integration
DeepLearning4J – Deep learning framework
ND4J – Scientific computation for the JVM
GPU support with CUDA
CPU with/without Intel MKL

Building and training a FAQ model

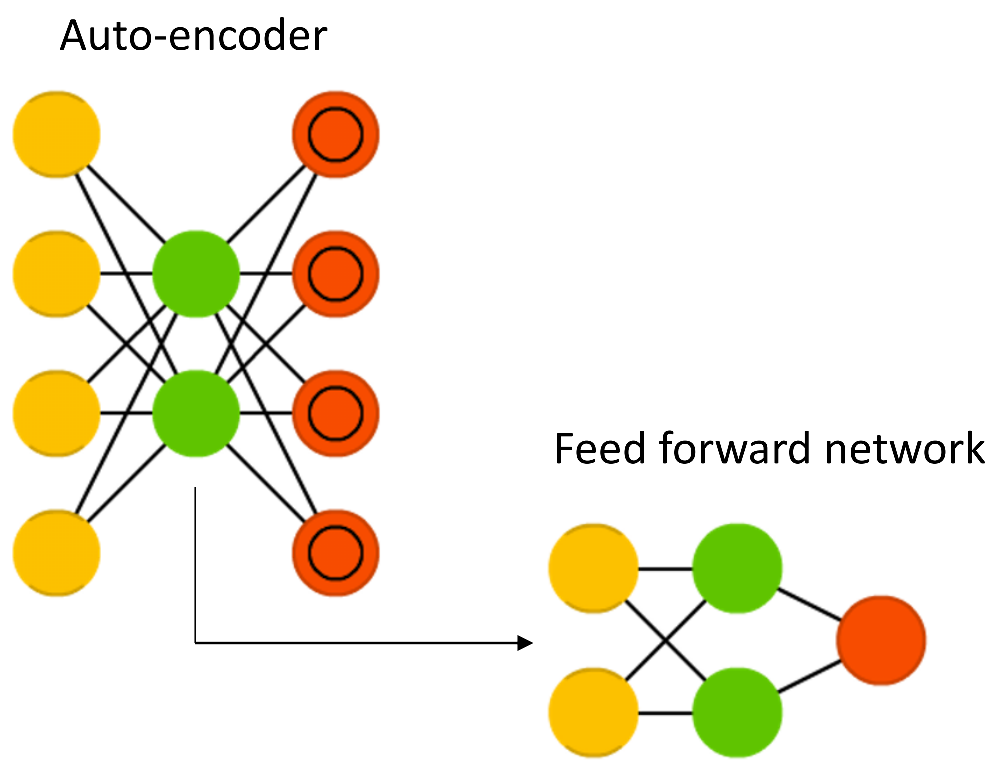
- Step 1: Build the neural network
- Step 2: Encode the input and output
- Step 3: Train the neural network

Step 1: Build the neural network

Fingerprint the data with an auto-encoder



Relate the fingerprint to an answer



```java
MultiLayerConfiguration networkConfiguration = new NeuralNetConfiguration.Builder()
  .seed(1337)
  .list()
    .layer(0, new VariationalAutoencoder.Builder()
      .nIn(inputLayerSize).nOut(1024)
      .encoderLayerSizes(1024, 512, 256, 128)
      .decoderLayerSizes(128, 256, 512, 1024)
      .lossFunction(Activation.RELU, LossFunctions.LossFunction.MSE)
      .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
      .dropOut(0.8)
      .build())
    .layer(1, new OutputLayer.Builder()
      .nIn(1024).nOut(outputLayerSize)
      .activation(Activation.SOFTMAX)
      .lossFunction(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
      .build())
  .updater(new RmsProp(0.01))
  .pretrain(true)
  .backprop(true)
  .build();
```

```java
MultiLayerConfiguration networkConfiguration = new NeuralNetConfiguration.Builder()
  .seed(1337)
  .list()
    .layer(0, new VariationalAutoencoder.Builder()
      .nIn(inputLayerSize).nOut(1024)
      .encoderLayerSizes(1024, 512, 256, 128)
      .decoderLayerSizes(128, 256, 512, 1024)
      .lossFunction(Activation.RELU, LossFunctions.LossFunction.MSE)
      .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
      .dropOut(0.8)
      .build())
    .layer(1, new OutputLayer.Builder()
      .nIn(1024).nOut(outputLayerSize)
      .activation(Activation.SOFTMAX)
      .lossFunction(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
      .build())
  .updater(new RmsProp(0.01))
  .pretrain(true)
  .backprop(true)
  .build();
```

```java
MultiLayerConfiguration networkConfiguration = new NeuralNetConfiguration.Builder()
  .seed(1337)
  .list()
    .layer(0, new VariationalAutoencoder.Builder()
      .nIn(inputLayerSize).nOut(1024)
      .encoderLayerSizes(1024, 512, 256, 128)
      .decoderLayerSizes(128, 256, 512, 1024)
      .lossFunction(Activation.RELU, LossFunctions.LossFunction.MSE)
      .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
      .dropOut(0.8)
      .build())
    .layer(1, new OutputLayer.Builder()
      .nIn(1024).nOut(outputLayerSize)
      .activation(Activation.SOFTMAX)
      .lossFunction(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
      .build())
  .updater(new RmsProp(0.01))
  .pretrain(true)
  .backprop(true)
  .build();
```

```
MultiLayerConfiguration networkConfiguration = new NeuralNetConfiguration.Builder()
  .seed(1337)
  .list()
    .layer(0, new VariationalAutoencoder.Builder()
      .nIn(inputLayerSize).nOut(1024)
      .encoderLayerSizes(1024, 512, 256, 128)
      .decoderLayerSizes(128, 256, 512, 1024)
      .lossFunction(Activation.RELU, LossFunctions.LossFunction.MSE)
      .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
      .dropOut(0.8)
      .build())
    .layer(1, new OutputLayer.Builder()
      .nIn(1024).nOut(outputLayerSize)
      .activation(Activation.SOFTMAX)
      .lossFunction(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
      .build())
  .updater(new RmsProp(0.01))
  .pretrain(true)
  .backprop(true)
  .build();
```

```
MultiLayerNetwork network = new MultiLayerNetwork(networkConfiguration);
network.setListeners(new ScoreIterationListener(1));
network.init();
```

Step 2:
Encode the input and output

Encoding text as a bag of words

Three steps:

1. Create a vector equal to the size of your vocabulary
2. Count word ocurrences
3. Assign the count each word a unique index in the vector

$$X_{train} = \left[\begin{array}{c} 0 \\ 1 \\ 1 \end{array}\right] \begin{array}{l} \\ \leftarrow \text{Hello} \\ \leftarrow \text{World} \end{array}$$

Create a bag of words in DL4J

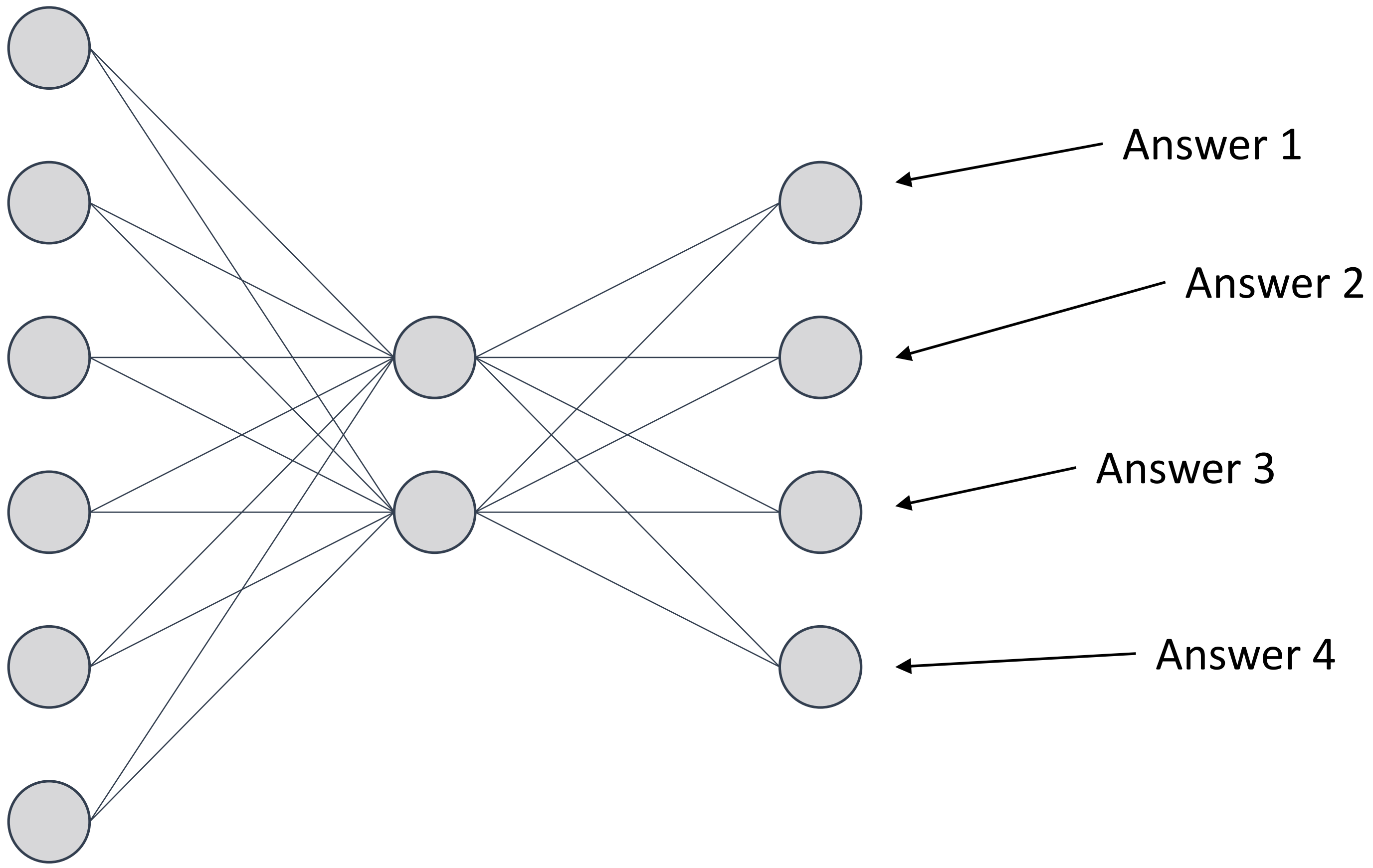
```
TokenizerFactory tokenizerFactory = new DefaultTokenizerFactory();
tokenizerFactory.setTokenPreProcessor(new CommonPreprocessor());
```

Create a bag of words in DL4J

```
TokenizerFactory tokenizerFactory = new DefaultTokenizerFactory();
tokenizerFactory.setTokenPreProcessor(new CommonPreprocessor());


BagOfWordsVectorizer vectorizer = new BagOfWordsVectorizer.Builder()
  .setTokenizerFactory(tokenizerFactory)
  .setIterator(new CSVSentenceIterator(inputFile))
  .build();
```

Encode answers



Map neurons to answers

```
try (CSVRecordReader reader = new CSVRecordReader(1, ',')) {
  reader.initialize(new FileSplit(inputFile));
}
```

Map neurons to answers

```
try (CSVRecordReader reader = new CSVRecordReader(1, ',')) {
  reader.initialize(new FileSplit(inputFile));

  Map<Integer, String> answers = new HashMap<>();

  while(reader.hasNext()) {
    List<Writable> record = reader.next();
    answers.put(record.get(0).toInt() - 1, record.get(1).toString());
  }

  return answers;
}
```

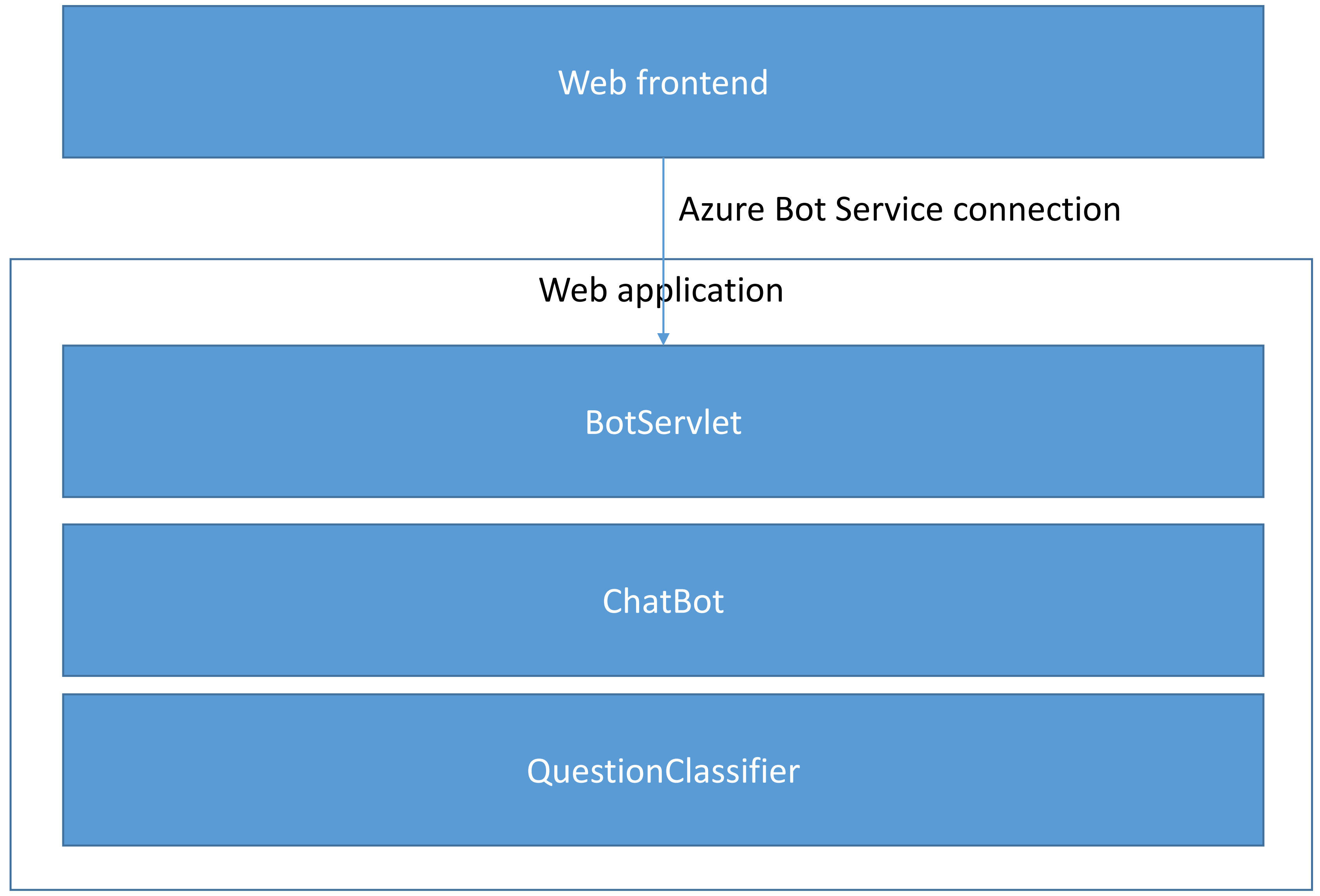
Step 3: Train the neural network

```java
QuestionDataSource dataSource = new QuestionDataSource(
        inputFile, vectorizer, 32, answers.size());

for (int epoch = 0; epoch < 100; epoch++) {
    while (dataSource.hasNext()) {
        Batch nextBatch = dataSource.next();
        network.fit(nextBatch.getFeatures(), nextBatch.getLabels());
    }

    dataSource.reset();
}
```

Using the neural network


Web frontend
Azure Bot Service connection
Web application
BotServlet
ChatBot
QuestionClassifier

Answering a question

Inside the bot framework adapter

```
String replyText = classifier.predict(context.activity().text());
```

At neural network level

```
INDArray prediction = network.output(vectorizer.transform(text));
int answerIndex = prediction.argMax(1).getInt(0,0);

return answers.get(answerIndex);
```

How to get started yourself

You too can use deep learning

- Three tips

  1. Explore the model zoo
  2. Starts with small experiments
  3. Choose a framework like DeepLearning4J

Useful resources

- The code:
  https://github.com/wmeints/qna-bot
- The model zoo:
  http://www.asimovinstitute.org/neural-network-zoo/
- DeepLearning4J website:
  http://deeplearning4j.org
- Machine learning simplified:
  https://www.youtube.com/watch?v=b99UVkWzYTQ&t=5s

Willem Meints

Technical Evangelist

@willem_meints
willem.meints@infosupport.com
www.linkedin.com/in/wmeints

Please

Remember to rate this session

Thank you!

@GOTOber

gotober.com