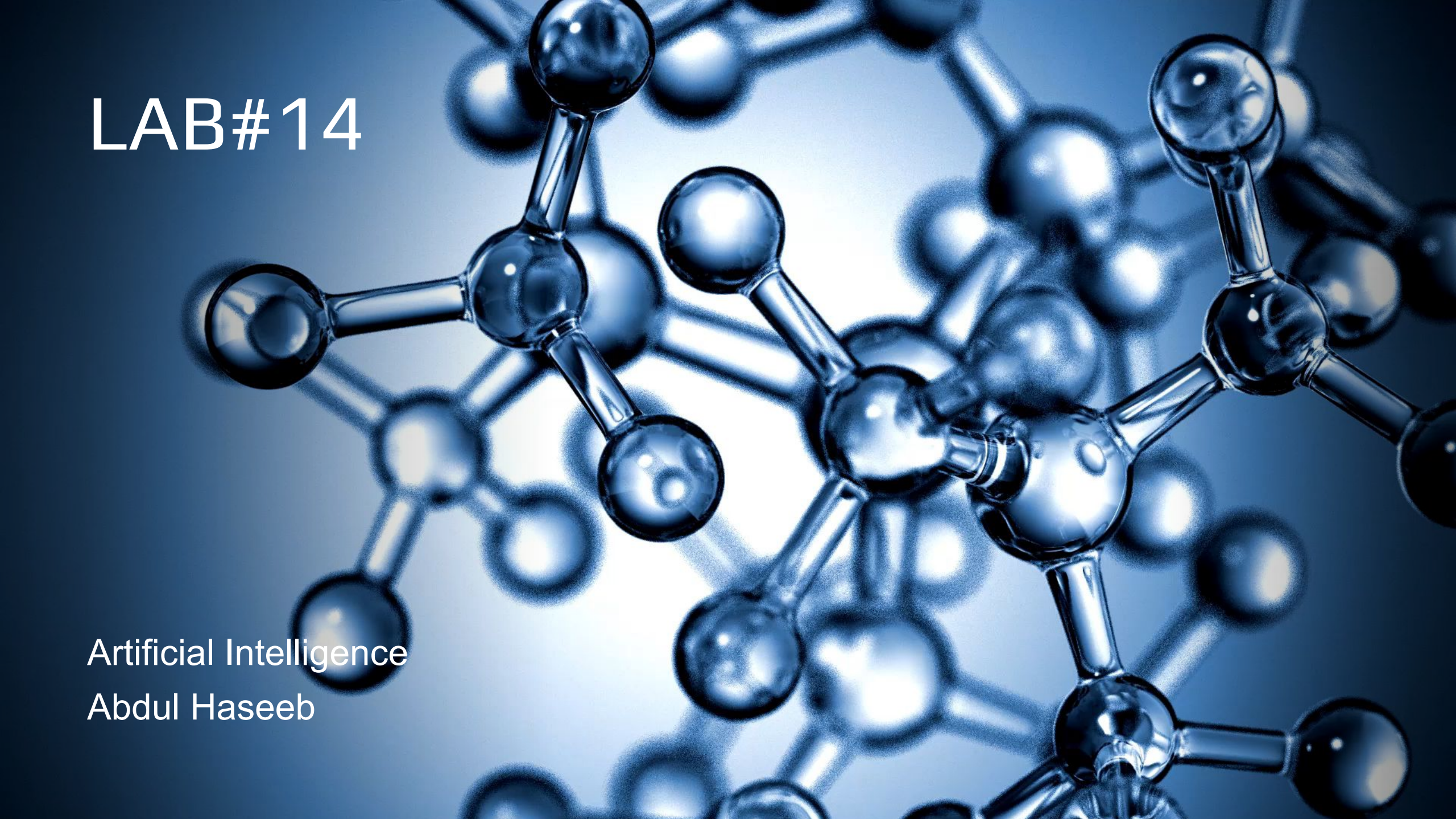
# LAB#14



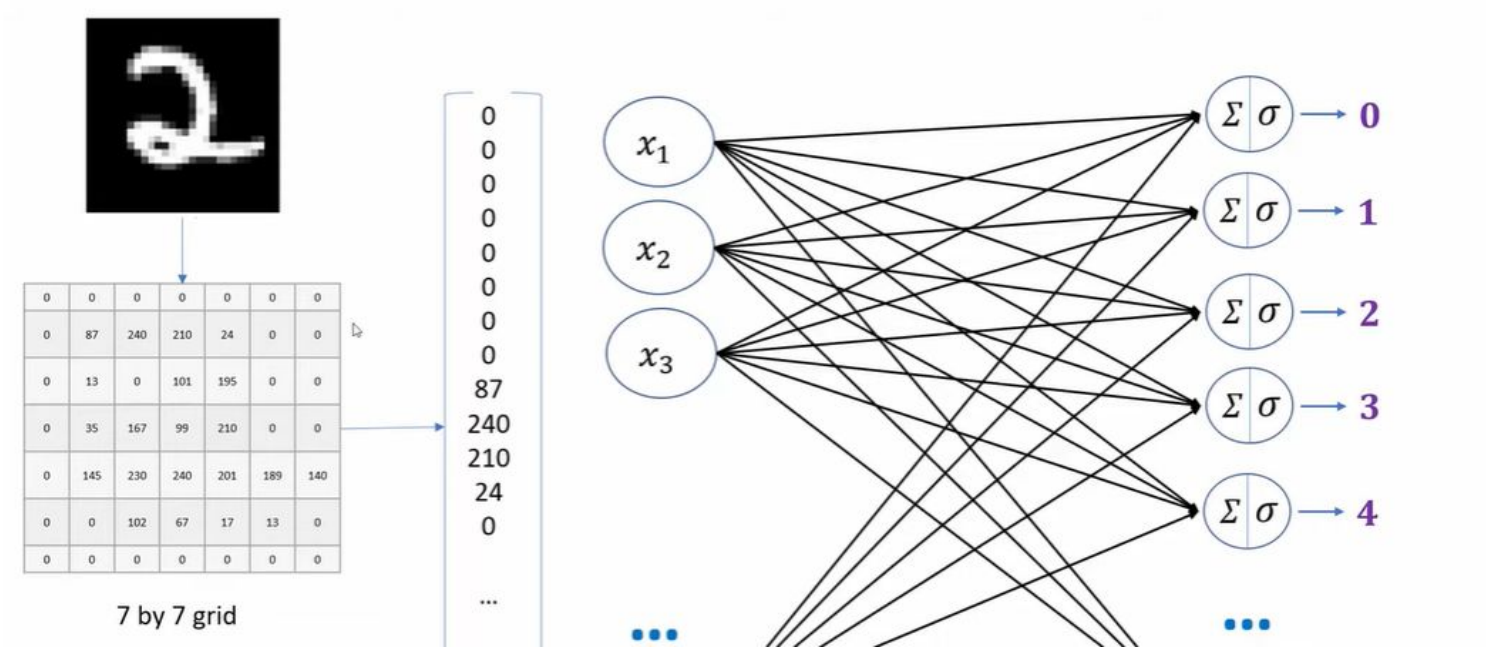
Artificial Intelligence

Abdul Haseeb

# Lab Objectives

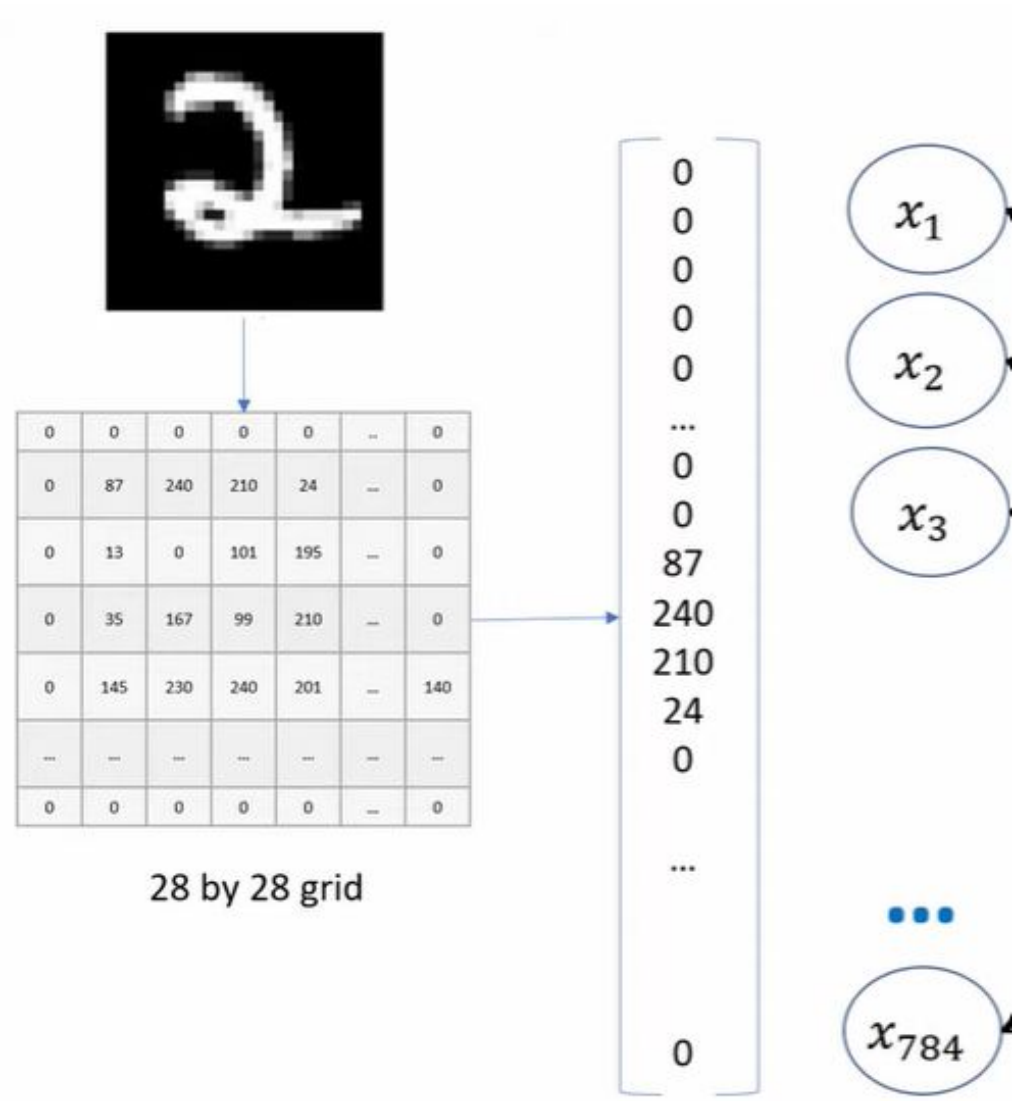
- Implementation of CNN on MNIST Dataset
- Implementation of LSTMs

# Passing Images to a neural Network



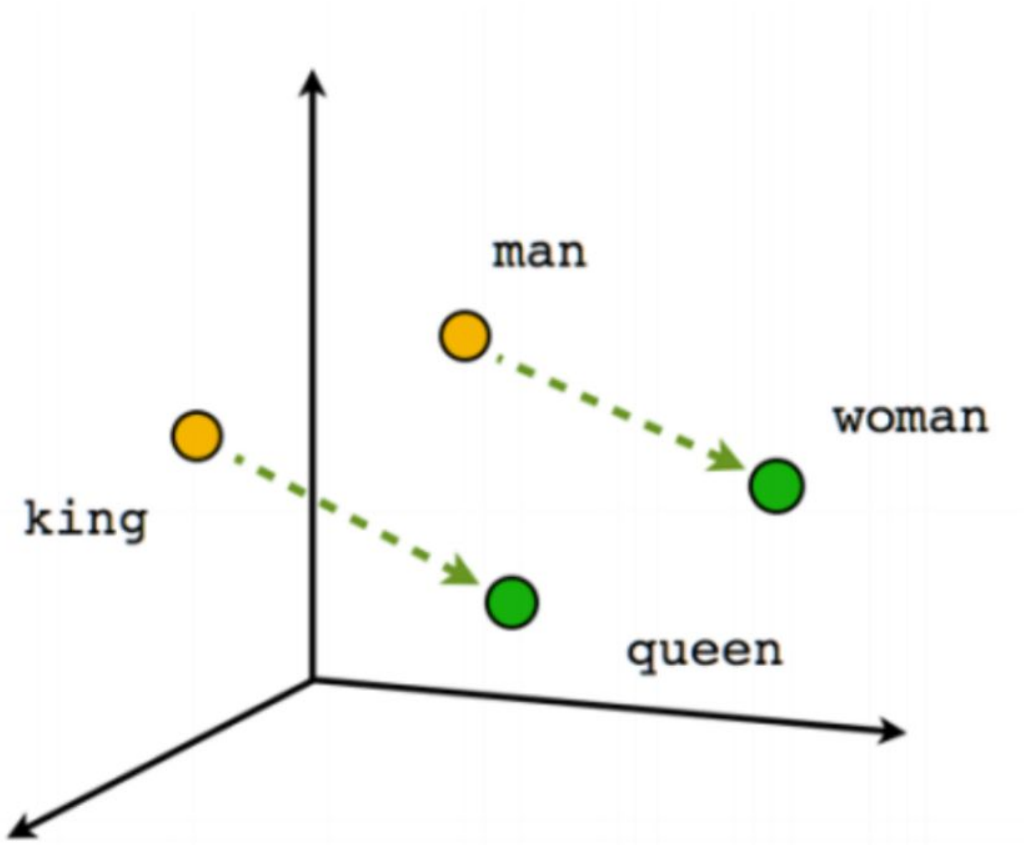
For a 28*28 Image we will have 784 neurons

# Passing Images to a network



# LSTMs

# Word Embeddings



```
text = 'Hi this is a small sentence'

# We choose a sequence length
seq_len = 3

# Split text into a list of words
words = text.split()
```

```
['Hi', 'this', 'is', 'a', 'small', 'sentence']
```

# First Step

```
# Make lines
lines = []
for i in range(seq_len, len(words) + 1):
  line = ' '.join(words[i-seq_len:i])
  lines.append(line)
```

```
['Hi this is', 'this is a', 'is a small', 'a small sentence']
```

# First Step(Cont..)

```
# Import Tokenizer from keras preprocessing text
from tensorflow.keras.preprocessing.text import Tokenizer
# Instantiate Tokenizer
tokenizer = Tokenizer()
# Fit it on the previous lines
tokenizer.fit_on_texts(lines)
# Turn the lines into numeric sequences
sequences = tokenizer.texts_to_sequences(lines)
```

```
array([[5, 3, 1], [3, 1, 2], [1, 2, 4], [2, 4, 6]])
```

```
print(tokenizer.index_word)
```

```
{1: 'is', 2: 'a', 3: 'this', 4: 'small', 5: 'hi', 6: 'sentence'}
```

# Second Step

```
# Import Dense, LSTM and Embedding layers
from tensorflow.keras.layers import Dense, LSTM, Embedding
model = Sequential()
# Vocabulary size
vocab_size = len(tokenizer.index_word) + 1
# Starting with an embedding layer
model.add(Embedding(input_dim=vocab_size, output_dim=8, input_length=2))
# Adding an LSTM layer
model.add(LSTM(8))

# Adding a Dense hidden layer
model.add(Dense(8, activation='relu'))
# Adding an output layer with softmax
model.add(Dense(vocab_size, activation='softmax'))
```

# Third Step