

Introduction to Deep Learning

IADS Summer-School 2022

Dr Haider Raza

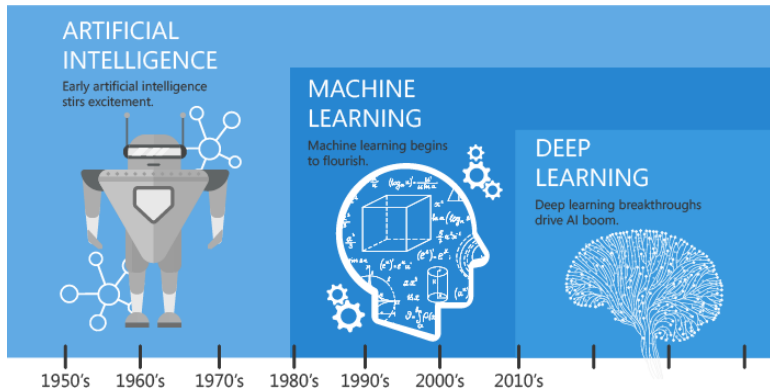
Lecturer in AI for Decision-Making
University of Essex July 26-27, 2022

What is Deep Learning

Deep Learning for Text

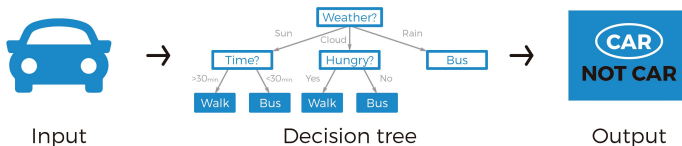
Conclusion

WHAT IS DEEP LEARNING?

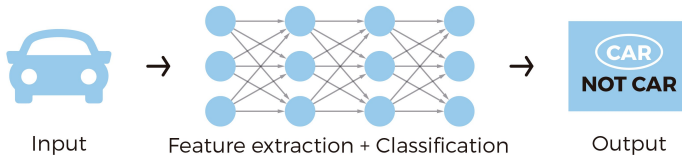


WHY IS DEEP LEARNING?

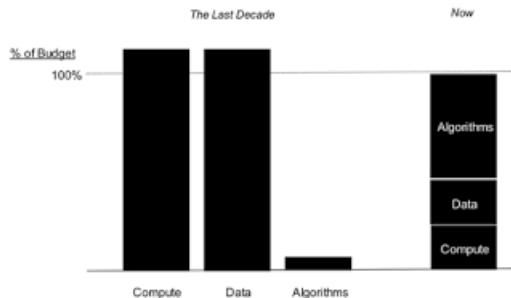
Machine Learning



Deep Learning

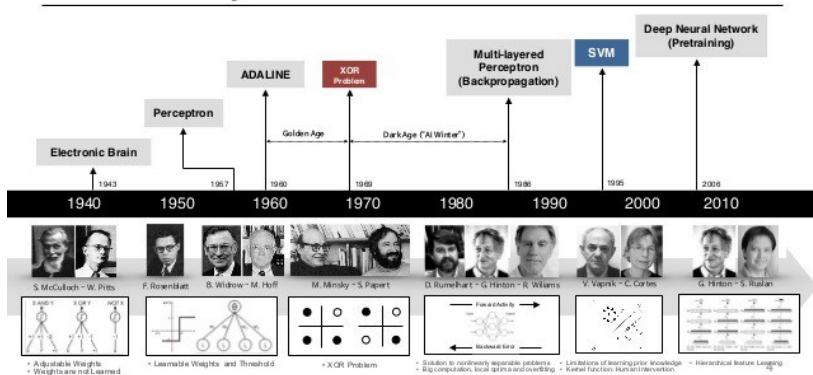


WHY IS DEEP LEARNING, NOW?



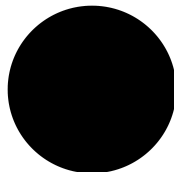
MACHINE LEARNING TO DEEP LEARNING TIME-LINE?

Brief History of Neural Network



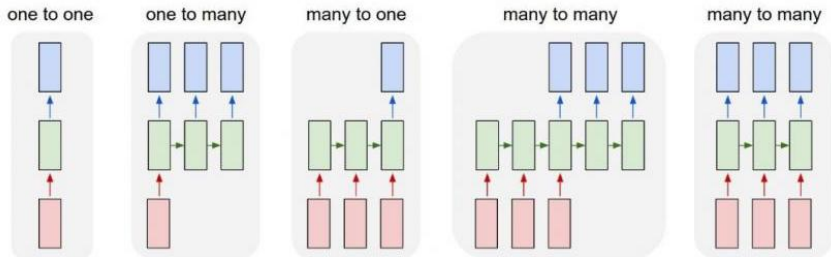
DEEP LEARNING FOR TEXT

Do you know where this circle will move next?



DEEP SEQUENCE MODELLING

Recurrent Networks offer a lot of flexibility:



↖ e.g. **Machine Translation**
seq of words → seq of words

WHAT IS SEQUENCE PROCESSING

- ▶ One-to-One: Binary Classification: Will I pass this class?: Yes/No
- ▶ Many-to-One: Sentiment Classification: Tweet Classification
- ▶ One-to-Many: Image Captioning: Show single-image to get the caption
- ▶ Many-to-Many: Machine Translation: English to any language such as French, Hindi, Urdu, German.

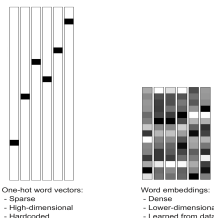
WHY RNN?

Unlike other algorithms, NN can also encode useful and obvious relationship in the data domain

1. Local spatial dependencies (computer vision: CNN)
2. Time dependencies (language and speech: RNN)

Note: Keep in mind throughout that none of deep-learning models truly understand text in a human sense; rather, these models can map the statistical structure of written language, which is sufficient to solve many simple textual tasks

WORD VECTOR VS WORD EMBEDDINGS



- ▶ the vectors obtained through one-hot encoding are binary, sparse (mostly made of zeros), and very high-dimensional (same dimensionality as the number of words in the vocabulary)
- ▶ word embeddings are low dimensional floating-point vectors (that is, dense vectors, as opposed to sparse vectors)
- ▶ So, word embeddings pack more information into far fewer dimensions

LSTM FOR IMDB REVIEW DATASET

```
from keras.layers import LSTM

model = Sequential()
model.add(Embedding(max_features, 32))
model.add(LSTM(32))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='rmsprop',
              loss='binary_crossentropy',
              metrics=['acc'])
history = model.fit(input_train, y_train,
                    epochs=10,
                    batch_size=128,
                    validation_split=0.2)
```

Achieve up to 89% validation accuracy. Not bad: certainly much better than the SimpleRNN network

CONCLUSION

- ▶ We have touched upon neural networks
- ▶ It's a really hot topic right now
- ▶ Requires a bit of dedication
- ▶ New algorithms are coming out every day