Machine Learning Concepts

_

End-to-End NLP model

1. Data Collection and Preparation

- Dataset: Start with your dataset comprising lines of text. Ensure it's properly labeled for classification.
- Data Cleaning: Clean the text data by removing irrelevant characters, correcting typos, and standardizing the text format.

2. Text Preprocessing

- **Tokenization**: Convert the text into tokens (e.g., words or characters).
- **Stopwords Removal (Optional)**: Remove common words that may not contribute to the meaning of the text.
- **Stemming/Lemmatization (Optional)**: Reduce words to their base or root form.
- **Vectorization**: Convert text tokens into numerical values. This can be done using techniques like Bag-of-Words, TF-IDF, or Word Embeddings.
- **Padding**: Since RNNs require inputs of the same length, pad shorter sequences with zeros or truncate longer ones to a fixed length.

3. Splitting the Dataset

- **Training and Test Set**: Split your dataset into a training set and a test set (commonly in an 80-20 ratio).
- **Validation Set (Optional)**: Consider creating a validation set or using techniques like cross-validation for model tuning.

4. Designing the RNN Model

- **Input Layer**: The input layer should match the dimensionality of the vectorized text data.
- **RNN Layers**: Add one or more RNN layers. You can use basic RNNs, LSTMs, or GRUs, depending on the need to capture long-term dependencies.
- **Dropout Layers (Optional)**: Include dropout layers to prevent overfitting.
- **Output Layer**: The output layer should correspond to the number of classes. Use a softmax activation function for multi-class classification.

- **Compile the Model**: Choose an appropriate optimizer (like Adam), a loss function (like categorical crossentropy for multi-class classification), and metrics (like accuracy).

5. Training the Model
- **Feed the Data**: Train the model using the training data. Set the number of epochs and batch size.
- **Validation**: Use the validation set (if available) to tune the hyperparameters and to prevent overfitting.

6. Model Evaluation
- **Test Set Evaluation**: After training, evaluate the model's performance on the test set to assess its generalization capability.
- **Metrics**: Measure the performance using metrics like accuracy, precision, recall, and F1-score.

7. Hyperparameter Tuning
- **Adjust Parameters:** Depending on the model's performance, you might need to adjust hyperparameters like learning rate, number of RNN units, number of layers, and dropout rate.

8. Model Deployment
- **Deployment:** Once satisfied with the model's performance, deploy it for real-world use or further testing.

9. Monitoring and Updating
- **Monitor Performance**: Continuously monitor the model's performance and gather user feedback.
- **Iterative Improvement**: Update the model periodically based on new data or changing requirements.

–

Internal Covariant shift: Caused due to the change in distributions of inputs in successive neural layers. Causes instability in the model and slowing down of process. Solved by **Batch Normalization.**

Activation function

1. Sigmoid : non-linearity/ output between 0 and 1/ vanishing gradient problem
2. Tanh: output in -1,1/ non-linearity/ applicable where negative data is important/ vanishing gradient problem/ computationally expensive/ unsuitable for only non-negative data
3. ReLu: non-linearity/ piecewise linear therefore, easier to updates weight/ Solves issue of vanishing gradient/ sparse activation of neurons/
4. Leaky ReLu: Solves dead neuron issue in Relu

Judging the model

k-fold cross validation

1. Advantages:
    1. Each observation is used for validation atleast once
    2. Robust testing of the model (reduced bias)
1. Disadvantages:
    1. Problematic when considering sequential data
    2. Validation and test data must be from the same distribution

train-test-validation: Splitting the dataset for training and validation and test from the same distribution

Bias - Variance Trade off

Bias: Error in prediction across training data
Caused when data underfits. Model is generalized to the training data and is undertrained
Ways to solve high bias:

1. Need more data points
2. Complexify the model
3. Train longer
4. increase dimensionality

Variance: Error in prediction across test/ validation data. Overfitting training data

1. regularization

2. Reduce dimensionality
3. early stopping
4. dropout

Confusion Matrix
A matrix consisting of true and false positives and negatives
Precision: TP/(TP + FP), exclusivity check; classifying a non-spam email as spam
Recall/ sensitivity: TP/(TP+FN), Safety check: classifying a disease case to a non-disease case
Accuracy
Selectivity: TN/TN+FP; opposite to precision; when false positivity is a concern

---

–

Optimization Methods

Batch Normalization: solves internal covariate shift
Advantages:
1. Faster learning
2. easier weight initialization
3. Allows us to use high learning rates

Disadvantages:
1. Not good for RNNs: does not consider temporal dependency
2. Different Calculation between train and test set

Grid Search
The grid search algorithm is a widely-used method for hyperparameter optimization in machine learning. It systematically searches through a specified subset of hyperparameters, evaluating a model for each combination of these parameters. The objective is to find the best combination of parameters that yields the most effective model performance, typically measured by a predefined metric such as accuracy, F1 score, or mean squared error.

How Grid Search Works:

1. **Define Parameter Grid:** You specify a "grid" of hyperparameter values. This grid is essentially a set of different values for each hyperparameter that you want to test.

2. **Cross-Validation:** For each combination of parameters in the grid, the algorithm trains a model using these parameters and then evaluates it using a cross-validation method. This typically involves splitting the dataset into multiple parts (folds), training the model on some folds, and validating it on the remaining folds, to ensure that the model's performance is assessed reliably.

3. **Model Evaluation:** After training on each combination of parameters, the model's performance is assessed using a predefined scoring metric.

4. **Select Best Parameters:** Once all combinations have been evaluated, the algorithm selects the combination that produced the best results according to the chosen metric.

Advantages of Grid Search:

- **Systematic Exploration:** It provides a thorough and systematic exploration of the parameter space, ensuring that the best combination within the grid is found.
- **Simplicity:** The approach is conceptually simple and easy to implement.
- **Parallelization:** Grid search can be parallelized easily since each combination of parameters can be evaluated independently of others.

Disadvantages of Grid Search:

- **Computational Cost:** The main disadvantage is its computational cost. The number of models that need to be trained and evaluated grows exponentially with the number of parameters and the number of values per parameter.
- **Limited Range:** It only searches through a predefined set of values, which means that the optimal set of parameters might not be included in the grid.
- **Resolution and Range Trade-off:** There's a trade-off between the granularity of the grid (resolution) and the range of parameter values. A finer grid (more values for each parameter) gives a more thorough search, but at the expense of greater computational cost.

_

Mini- Batch Gradient Descent (GD with small batches-weights updated for every epoch)

Advantages:

1. Efficient
2. Computationally less taxing
3. Faster at converging as weights are updated frequently

Disadvantages:

1. Risk of reaching local minima
2. inconsistent convergence
3. sensitive towards batch size
4. Dependence upon data-shuffling

Stochastic Gradient Descent

Advantages:

1. Efficient
2. Quick
3. No risk of reaching a local minima

Disadvantages:

1. Not a smooth convergence
2. Need of higher epochs

Gradient Descent

Momentum

Exponentially weighted average
Acceleration term speeds up momentum according to the slope at that time. If it is too high it will reduce/increase velocity by higher magnitudes and similarly for lower values, it does by smaller magnitudes
most common value for beta =0.9

Learning Rate Decay

Decreasing learning rate for every epoch
alpha = (1/(1 + decay* no. of epochs))*aplha0

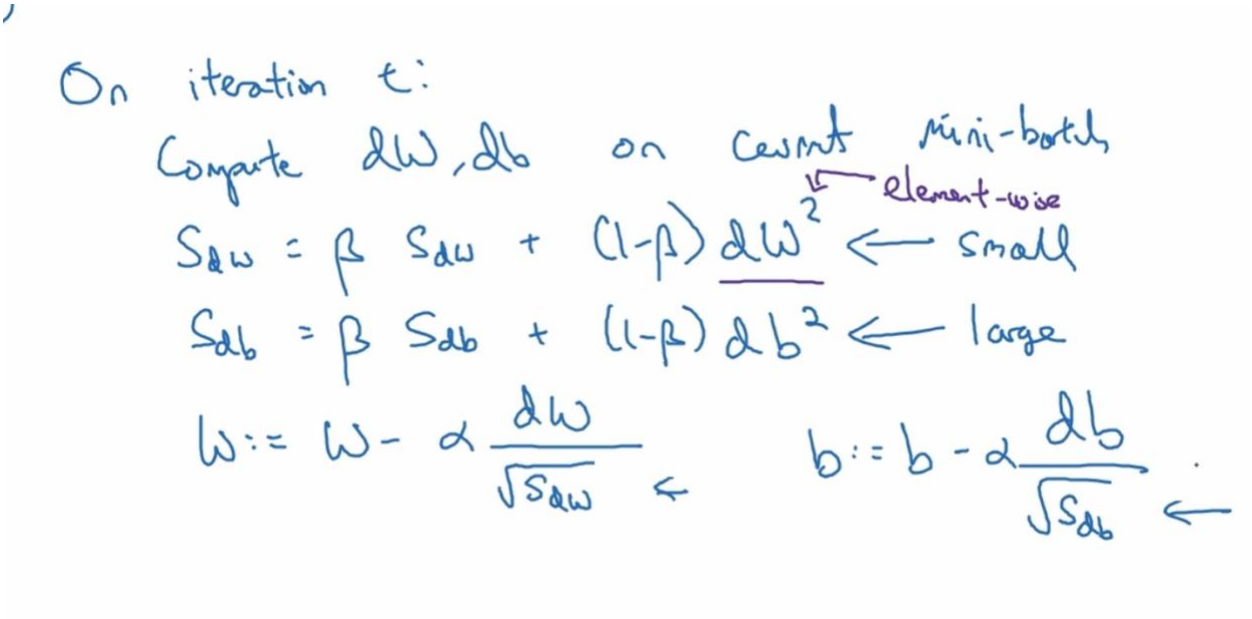
1. exponentially decay

2. constant/ (epochs)^1/2 decay

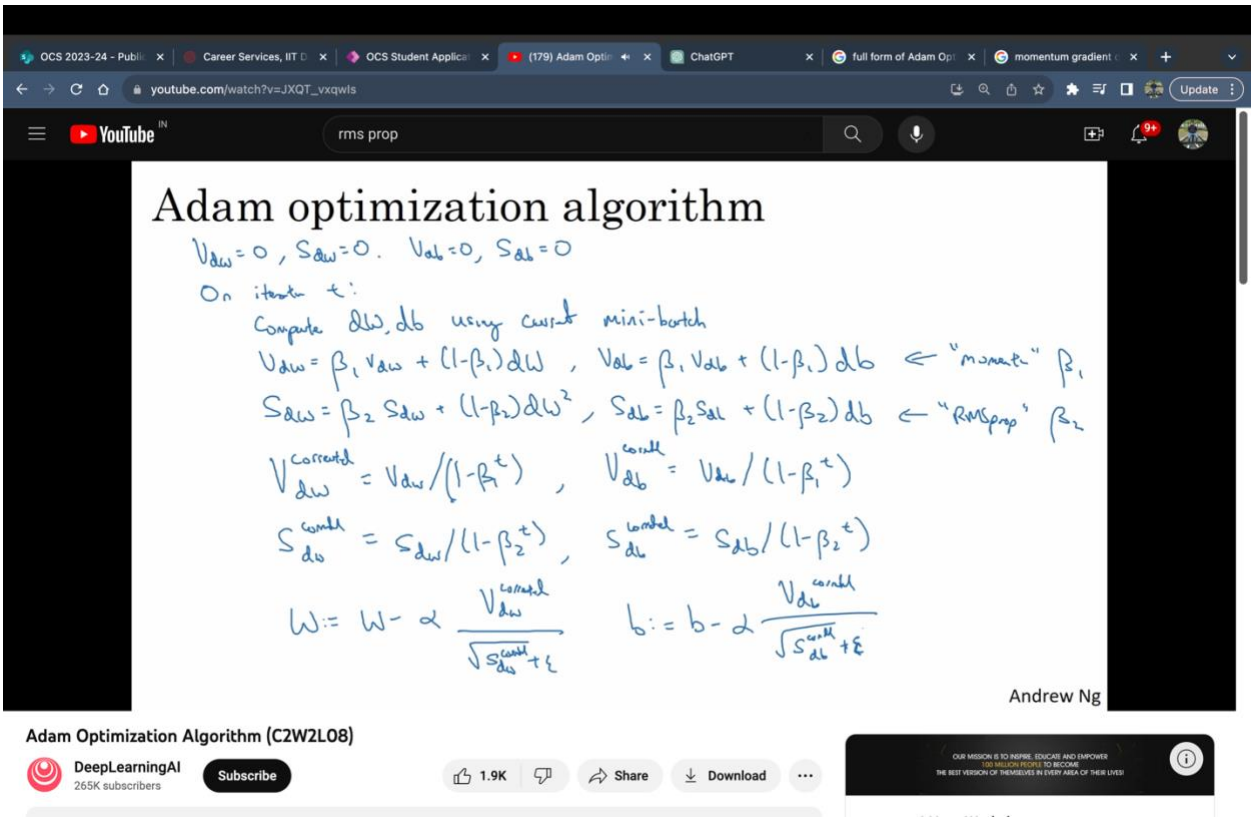
RMS Prop

Speeds up gradient descent

Slow down learning in vertical direction and fasten the learning in horizontal direction



Adam Optimization (Adaptive moment estimation)



Advantages:

1. Adaptive learning rates
2. automatic bias correction
3. quick convergence

Disadvantages:

1. aggressive learning rate can lead to divergence sometimes
2. Can reach sub-optimal solutions sometimes

Bagging vs Boosting

Bagging:

1. Parallel learning by dividing into subsets and each model ensemble learning on different subsets. Subsets are sampled
2. Reduces variance in model
3. Example: Random Forests
4. averaging out results across all models or by voting in classification
5. Effective in low bias- high variance systems

Boosting

1. Sequential learning of a single model. Each model learns the mistakes from the previous model
2. used to reduce bias

3. each iteration focuses on improving performance across weak learners
4. averaging out results across all models or by voting in classification across all previously exisitng models
5. Uses Pruning
6. Self Regularization
7. Parallelization is not possible

Loss functions

1. Mean Squared Error: high penalty to large errors
2. Mean Absolute Error: less sensitive to outliers
3. Cross Entropy Loss/ Binary Cross Entropy Loss
4. Categorical Cross - entropy
5. Hinge Loss
6. Huber Loss (MAE +MSE)

Machine Learning Models

LGBM

Model Architecture:
Primary Application: regression and classification
Advantages:

5. Faster than XGBoost. Finds more efficient algorithm to find the optimal split points for each tree
6. More memory efficient that XGB as stores data in compressed form. Helpful for large datasets
7. Handles categorical features without the need of one-hot encoding them. Uses Fisher method

Disadvantages:

2. Overfits on small models
3. Tuning is very complex due to the incorporation of several hyper-parameters
4. Requires high computational resources to work on large datasets

Applications:

5. Regression and classification tasks
6. Large dataset low accuracy
7. Data with a lot of categorical features

Special features:

5. Gradient based one sided sampling (GOSS)
6. Feature Bundling (boosts speed)
7. Special algorithm for categorical features
8. Good support for parallel and GPU learning

GOSS:

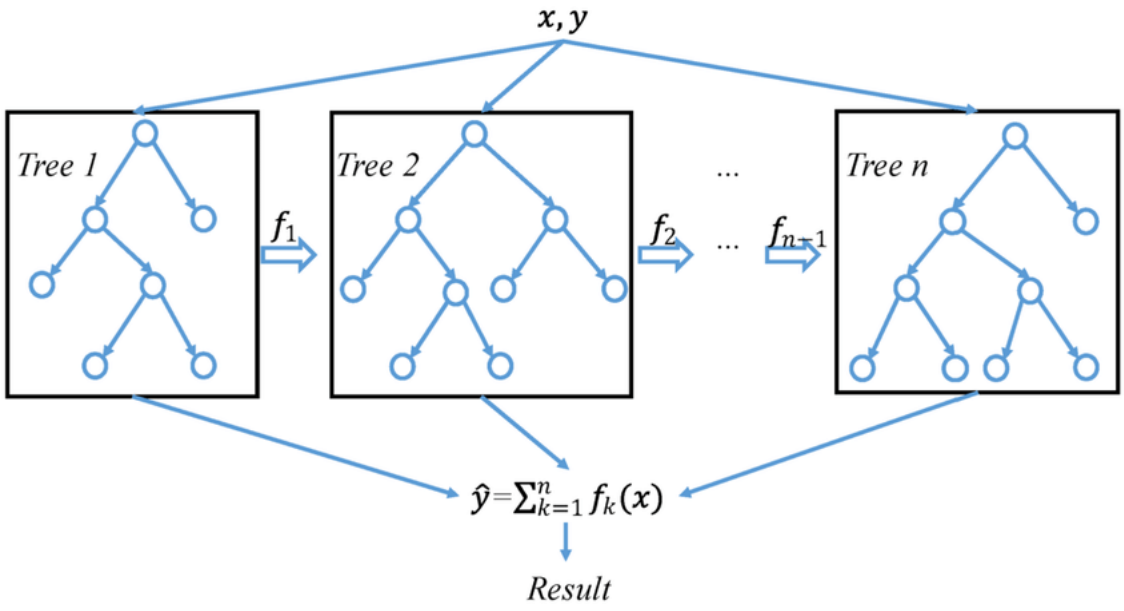
4. Focuses on learning on large errors by sampling large error data points at a higher rate
5. Smaller weights assigned to sampling of data points with small error

Benchmark set against: **XGBM**

–

XGBM

Model Architecture:



Primary Application:

3. High accuracy, low efficiency tasks
4. Small datasets

Advantages:

4. Robustness due to extreme boosting. high accuracy as well, robust overfitting
5. tree pruning: takes trees to maximum depth and removes all splits which dont help
6. Can be used for classification, regression, ranking, and user-defined prediction problems
7. Has a regularization element
8. Good at handling missing data

Disadvantages:

5. Prone to overfit
6. Complex model hence computationally intensive
7. Cannot preprocess categorical features
8. Not the best model for multi-class classification:

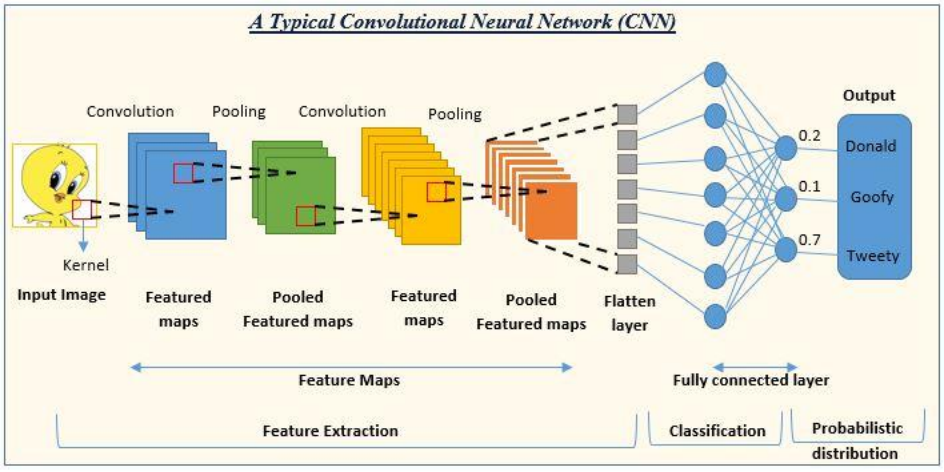
Special Features:

4. Pruning action
5. Regularization already present

Benchmark Set against:Decision trees, random forests

CNNs

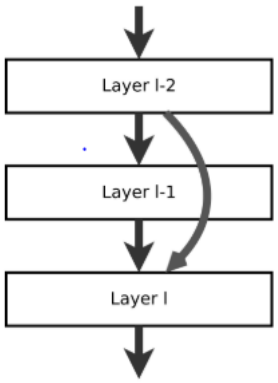
1. primary application: classificaton/ identification/ localization tasks
2. model architecture:



3. advantages:
3. Local connectivity
4. Shared Weights: hyper-parameters belong to filter
5. automatic feature learning
6. Generally invariant to small transformations, distortions, and translations in the input image.
7. Hierarchical Feature Learning
4. disadvantages:
3. Black box nature: as we don't know how it learns. leading to challenges in understanding model decisions
4. overfitting: prone to overfitting on small datasets
5. Computationally intense/ complex architecture
5. which previous model does it succeed Sliding window techniques/ NNs
6. applications:
4. Image segmentation
5. classification
6. localizaton
7. visual perception tasks
7. why is it special?
3. pooling reduces dimensionalities
4. automatic feature hierarchy
5. shared weights

ResNet50

1. primary application: image classification on imageNet 1000
2. model architecture:



3. advantages:

6. ResNet50 introduces residual blocks with skip connections, allowing gradients to flow through the network directly

4. disadvantages:

8. Very big architecture (50 layers), hence computationally challenging

9. Highly dependent upon batch normalization

5. which previous model does it overceed: alexNet, CNN, VCGNet

6. special features: residual blocks solving skip connections

---

WideResNet

Main feature: Just like resnet but is wider rather than being deeper

Advantages: Less complex, easier to train

Disdavantages: prone to overfit

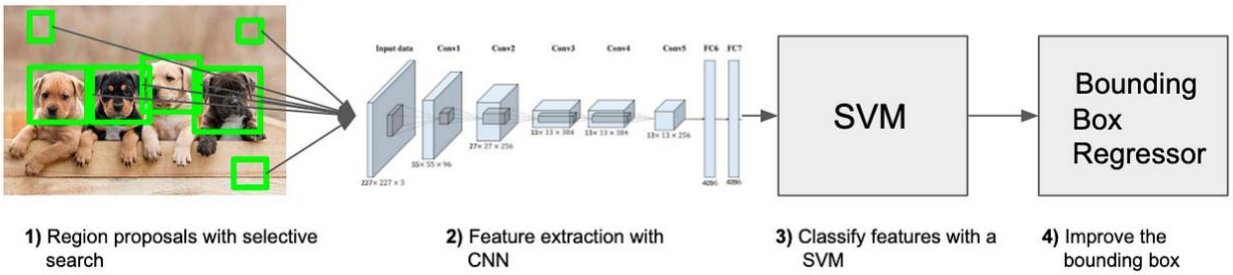
DenseNet

Main features: outputs of all feature maps are concatenated to all successive conv layers

RCNNs

1. primary application: image localization

2. model architecture:



3. advantages:

7. very good for image localization
8. faster than sliding window technique as it gives the particular region proposals
9. greater feature learning
10. has pre-trained AlexNet, hence, classification model is pre-trained

4. disadvantages:

1. Each region is processed individually hence very high computational time
2. Region proposals are inefficient
3. Multiple training stages: training the feature extracting CNNs, training the classifiers
4. slow real time object detection
5. Resizing every image is computationally heavy

5. which previous model does it overceed: Sliding Window technique + CNNs

6. special features: Integration of region proposals with CNNs

Output of RCNNs is a confidence score. How confident is that region consists this class.

n - no. of classes

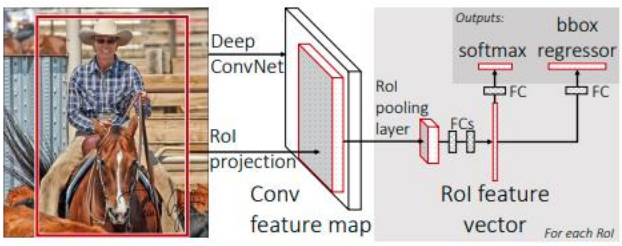
r - no. of region proposals

hence output is n*r matrix

Fast RCNNs

Primary application: Image localization

Model Architecture:



Advantages:

1. 9 times faster than RCNN
2. Loss is only calculated for positive samples

Disadvantages:

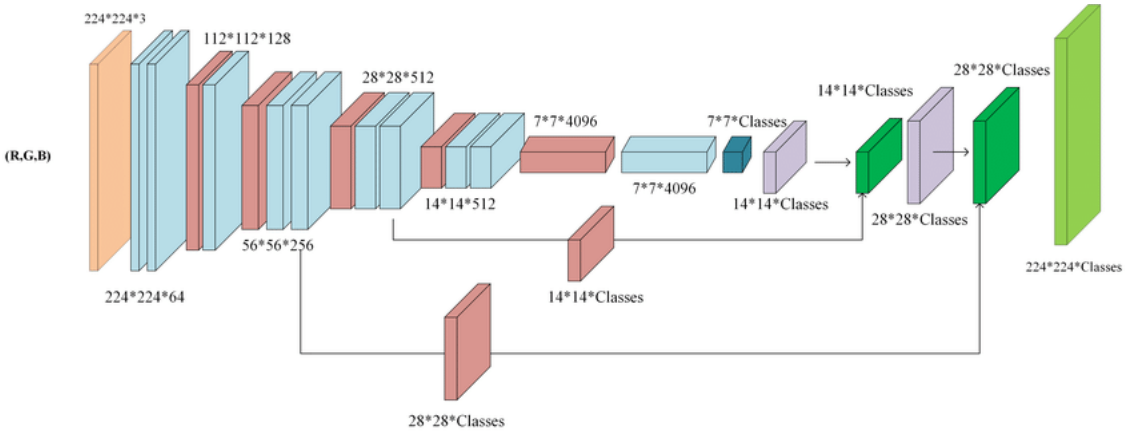
1. Depends upon external region proposal algorithms
2. Still not good for real-time processing

Special feature: ROI Pooling

FCN (Fully Convolutional Networks)

Primary task: Image Segmentation

Model Architecture:



Advantages:

1. Pixel wise segmentation
2. Shared Weights

Disadvantages:

1. imperfect boundaries
2. Fails to segment small objects

Special features: Encoder-Decoder Network

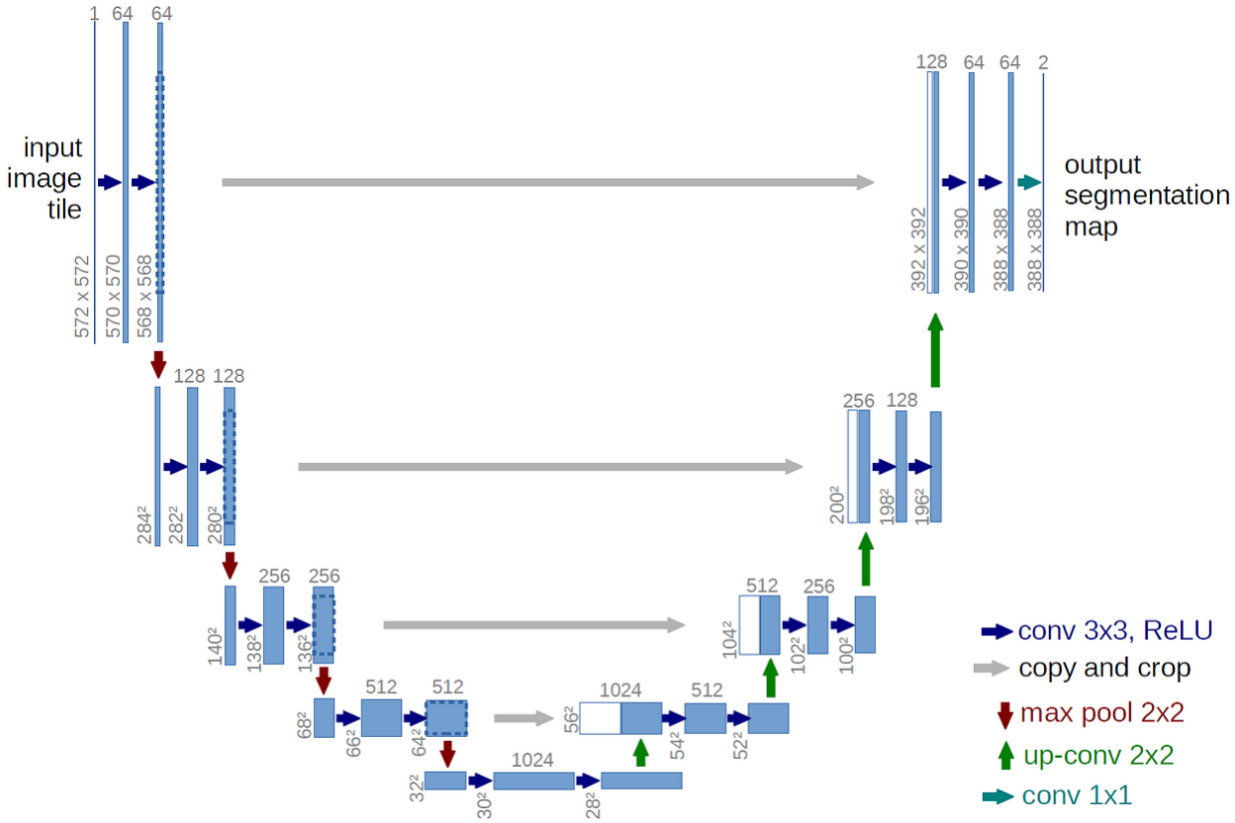
Applications:

1. Biomedical imaging
2. Autonomous Driving
3. Facial Recognition
4. Document Analysis
5. Traffic Control Systems

UNet

Primary Application: Image segmentation

Model Architecture:



Advantages:

1. Emperically found, UNet gives good results for smaller datasets
2. The skip connections in U-Net concatenate feature maps from the contracting path to the expansive path, improving the flow of information through the network.
3. Precise localization

Disadvantages:

1. Overfitting on small datasets
2. The receptive field of U-Net may sometimes be insufficient to capture larger context, which can be a limitation for certain segmentation tasks where global information is crucial.
3. Complex Architecture

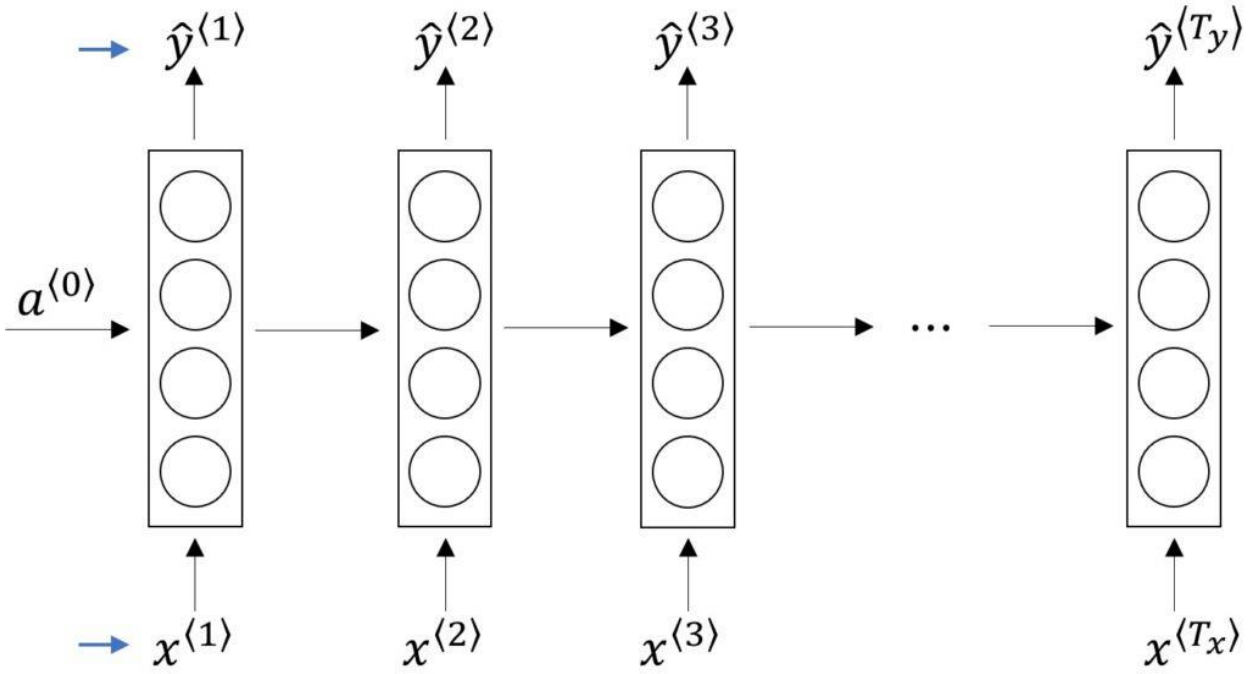
Special features:

1. The encoder network (contracting path) half the spatial dimensions and double the number of filters (feature channels) at each encoder block. Likewise, the decoder network doubles the spatial dimensions and half the number of feature channels.
2. These skip connections provide additional information that helps the decoder to generate better semantic features. They also act as a shortcut connection that helps the indirect flow of gradients to the earlier layers without any degradation.
3. The decoder block starts with a 2x2 transpose convolution. Next, it is concatenated with the corresponding skip connection feature map from the encoder block. These skip connections provide features from earlier layers that are sometimes lost due to the depth of the network.
4. The output of the last decoder passes through a 1x1 convolution with sigmoid activation. The sigmoid activation function gives the segmentation mask representing the pixel-wise classification.

benchmark set against: FCN, SegNet
Applications: Segmentation tasks

RNNs (Vanilla)

1. primary application: Handling sequential data
2. model architecture:



Many-to-many $T_x = T_y$

3. advantages:

1. Helpful in handling sequential data
2. Captures temporal information
3. Can handle data of varying space
4. used for real-time processing

4. Disadvantages:

1. Vanishing Gradient Problem
2. Exploding gradient problem
3. Short-term memory
4. difficult to train due to initial conditions and parameters

5. which previous model does it succeed: Neural Networks

6. special features: Shared weights

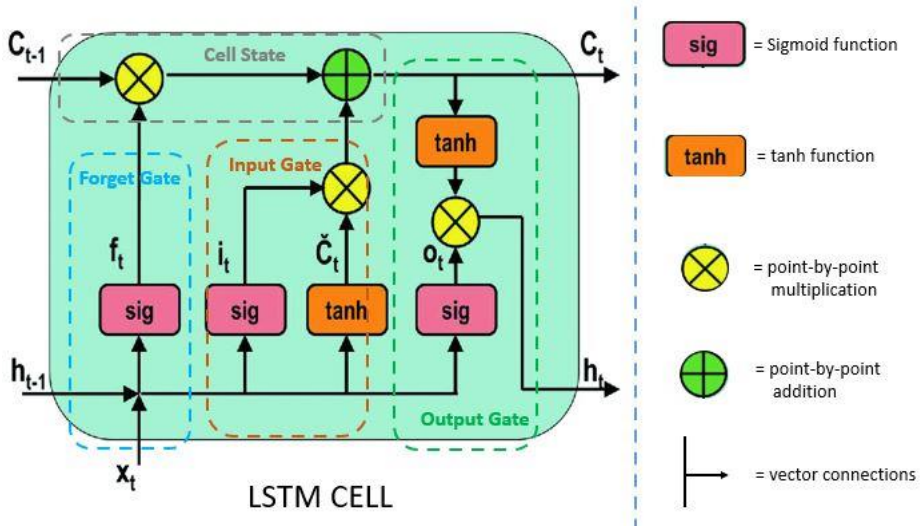
Applications:

1. financial forecasting
2. text translation
3. text-to-image conversion

–

LSTMs

primary application: sequential data
model architecture:



advantages:

1. remembers input data for longer periods
2. captures long-term dependencies
3. reduced risk of vanishing/ exploding gradients

speciality:

- **Forget Gate**: Decides what information is discarded from the cell state. It looks at the previous hidden state and the current input, and outputs a number between 0 and 1 for each number in the cell state. A 1 means “completely keep this” while a 0 means “completely get rid of this.”
- **Input Gate**: Decides what new information is stored in the cell state. It involves two parts: a sigmoid layer that decides which values to update, and a tanh layer that creates a vector of new candidate values that could be added to the state.
- **Output Gate**: Decides what the next hidden state should be. The hidden state contains information about previous inputs. The hidden state is also used for predictions.

applications: speech recognition, language modelling, translation
disadvantages:

1. prone to overfit
2. complex architecture

benchmarking above: RNNs

_

Transformers

Advantages:
1. Faster
2. Bidirectional

Disadvantages:
1. higher training time
2. Computationally expensive

BERT (Bidirectional Encoder Representation from transformers)

Applications:
1. Sentiment Analysis
2. Question Answering
3. Neural Machine Translation
4. Text Summarization

Advantages:
1. Already pre-trained on large datasets
2. Bidirectional Encoding
3. Just need a bit of fine-tuning to work on our dataset