what is data science:

data science is applying methods like statics,maths or artificial intelligence so we can extract valuable information , uncover insights and make predictions from the data we do have.

Deep Learning:

Neural network :

cat or not example :

Input Layer (16 Features): Imagine we have an input layer that represents an image as a 4x4 pixel grid, which gives us 16 features. Each feature corresponds to a specific pixel's intensity or color.
2. Hidden Layers (Feature Detection): We have a set of hidden layers responsible for feature detection. In this case, you've defined four neurons in this layer, each with its own set of weights. These neurons are trained to detect specific features within the image:

- Neuron 1 (Eyes Detector): This neuron specializes in detecting the presence of eyes in the image. It learns to recognize eye-like patterns or arrangements of pixels.
- Neuron 2 (Nose Detector): Neuron 2 is dedicated to identifying the presence of a nose. It looks for nose-like structures or patterns.
- Neuron 3 (Tail Detector): This neuron focuses on detecting the tail. It learns to recognize tail-related patterns.
- Neuron 4 (Ears Detector): Neuron 4's role is to identify ears in the image. It learns to recognize ear-like features.

Each of these neurons applies an activation function (such as ReLU) to the weighted sum of its inputs, introducing non-linearity.
3. Training and Weight Adjustment: During the training process, the neural network is provided with a labeled dataset of images, indicating whether each image contains a cat or not. The network's weights are adjusted through backpropagation and optimization algorithms like gradient descent to minimize a loss function. The training process helps the neurons specialize in detecting the corresponding features, making them sensitive to the presence or absence of eyes, nose, tail, and ears.
4. Final Layer (2 Neurons): The final layer is designed for classification and contains two neurons:

- Neuron 0 (Not a Cat): This neuron is trained to provide an output indicating the probability that the image does not contain a cat.
- Neuron 1 (Cat): Neuron 1 is trained to provide an output indicating the probability that the image contains a cat.

The activation function in the final layer is often softmax, which converts the raw outputs into probabilities. The class (cat or not cat) with the highest probability is considered the network's prediction

what is a neuron,perceptron :

neuron takes inputs multiply them by their optimized weights and then pass them through an activation function

activation function:

decides if a neuron should be activated or not.

relu: [0,+infinit] return the exact input value if it’s positive otherwise 0 is attributed
sigmoid:[0,1]
tanh:[-1,1]
softmax:[0,1]

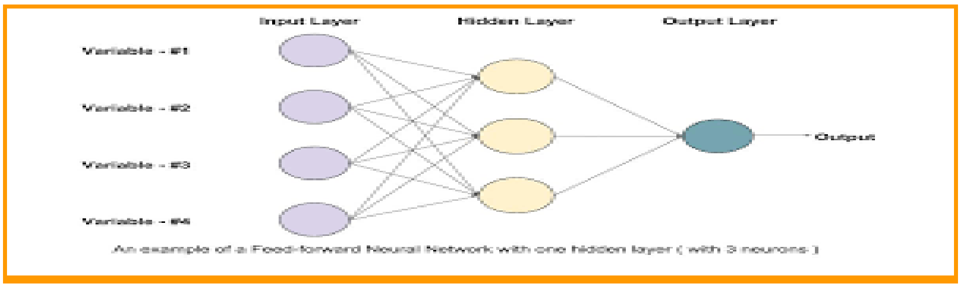
- Softmax is typically used in multi-class classification problems, where an input can belong to one of several mutually exclusive classes.
- Sigmoid is often used in binary classification problems, where the output can belong to one of two classes

Backpropagation or backward :

Backpropagationa a global optimization process works to optimize the weights and biases for all neurons in all layers so that the network learns to make better predictions,

Feed forward neural network: (regression and classifcation)MLP multippreceptron

> the ffnn doesn’t handle sequential data
interconnected layers , utilizes the backward propagation technique
typically used for classification and regression



An example of a Feed-forward Neural Network with one hidden layer (with 3 neurons)

##RNN:

works for the sequential data
if the input is sentence the tokens need to be encoded using word embedding model like word2vec

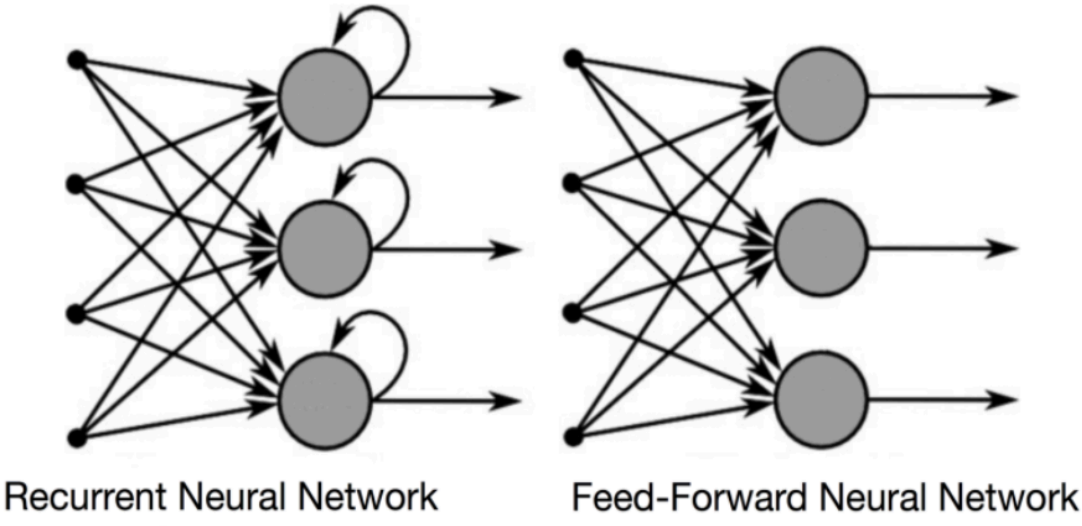
Recurrent Neural Networks (RNNs) are a subset of neural network architectures specifically designed for handling sequential data. Unlike feedforward neural networks, RNNs have a unique feature where the output of each neuron is fed back into itself at the next time step, creating a feedback loop. so each neuron has a memory called hidden state.
however Rnn suffer from two main problems:

vanishing : when the weights assigned are too small , so the gradient can’t convert to local minimum because it’s taking very small steps so it disappears .

exploding: when the weights assigned are too large, so the gradient can’t convert to the local minimum because it’s taking a large step so it can’t converge to the local minimum.

LSTM: to solve the vanishing and exploding problem

the lstm models are based on the rnn architecture but the neurons do not only includes hidden state but also cell state to maintain the long sequences and 3 gates:
1.forget gate :Decides which information should be droped from the cell state.
2.input gate: Decides which new information should be added to the cell state.
3. output gate : Decides what the next hidden state hth_tht should be based on the cell state.
 the cell state contains the input gate + the forget gate



What Is the Difference Between a Feedforward Neural Network and Recurrent Neural Network?

A Feedforward Neural Network signals travel in one direction from input to output. There are no feedback loops; the network considers only the current input. It cannot memorize previous inputs
A Recurrent Neural Network's signals travel in both directions, creating a looped network. It considers the current input with the previously received inputs for generating the output of a layer and can memorize past data due to its internal memory

CNN:
It is designed to extract features from image by identifying local patterns such as the edges, corners and then combine them to recognize the image. the output of the convolution neural network is passed through a pooling layer to reduce the size of the features.

pooling : It performs down-sampling operations to reduce the dimensionality and creates a pooled feature map by sliding a filter matrix over the input matrix.

flatten:taking two dimensional array and transform it ino one dimensional array

Four layers in CNN:

1. **Convolutional Layer -**It is designed to extract features from image by identifying local patterns such as the edges, corners and then combine them to recognize the image.
2. **ReLU Layer** - it brings non-linearity to the network and converts all the negative pixels to zero. The output is a rectified feature map.
3. **Pooling Layer** - pooling is a down-sampling operation that reduces the dimensionality of the feature map.
4. **Fully Connected Layer** - this layer recognizes and classifies the objects in the image.

What Are Hyperparameters:

A hyperparameter is a parameter whose value is set before the learning process begins such as the number of hidden units, the learning rate, epochs, etc

Batch normalization:

Batch normalization normalizes the outputs of a layer (which can be considered the hidden layer activations).

Dropout technique:

Dropout is a regularization technique used to prevent overfitting in neural networks by randomly "dropping out" (i.e., setting to zero) a fraction of the neurons during training.

stochastic gradient descent:

sgd process starts by taking iterativily one data point then calculates the error of that one and adjusts the weights accordingly.

batch gradient descent:

Batch gradient descent takes the whole training data calculate the cumulative errors and then adjust the weights

mini-batch gradient descent:

Mini batch gradient descent is similar to SGD but instead of taking one data sample it takes a random sample of data and then calculates the cumulative error and adjust the weights.

momentum:

SGD updates the weights for each random data point iteratively, which leads to high noise in the weight updates. To address this, momentum is introduced. Momentum is a value between 0 and 1 that helps to reduce noise and accelerate convergence. It keeps track of an additional term, which is a moving average of past gradients. The momentum term combines a fraction of the previous momentum and the current gradient, and the weights are updated according to the previous weights plus the momentum term..

RMSProp:

RMSProp (Root Mean Square Propagation) modifies the learning rate by dividing it by a term that accounts for the magnitude of recent gradients. RMSProp maintains a moving average of the squared gradients, which helps to adapt the learning rate for each parameter individually. This approach helps to maintain a stable learning rate and prevents the learning rate from becoming too large, which can lead to unstable training.

Adagrad:

Adam optimizer:

Tensor Flow

What Do You Mean by Tensor in Tensorflow?

It is an array but designed for higher dimensions.

Tensorboard:

It is a visualization tool for example to visualize the learning curve of a model , the loss function during training.

Types of tensors:

variable: mutable
constant:immutable
placeholders:It is used to assign data at a later point in time.

What are the methods that can be used to handle overfitting in TensorFlow?

- Batch normalization
- Regularization technique
- Dropouts

how to load data using tensorflow?

tensorflow load into memory or datasetapi piepline

Machine learning:

explain machine learning for someone who doesn't heard the term before:

okay imagine u are student and you are preparing for a big exam , so you have taken alot of exercises which in machine learning context called data and ofcourse you do have a teacher that corrects your mistakes and train you which is the optimization algorithm for your exam which is the model testing.
exercises=data
teacher=optimization model
exam=model testing

Gradient descent:

gradient descent is an optimization model used to update the weights of model in order to minimize the loss function.
The algorithm start with random initial parameters then iteratively update them according to step size and learning rate.
The process is repeated until the step size converge to zero or the number of epochs is reached.

loss function = The difference between the predicted output and the true value: residual

Grid search cv

To optimize the model hyperparameters

This process involves creating a grid of hyperparameter values and training and evaluating the model for each combination using cross-validation.

cross validation to evaluate the model

The algorithm split the data into batches and then train them and evaluate them to obtain several estimations of the model performance.
The validation score represents how well the model performs on unseen data .
The validation score can help you evaluate the learning curve of a specific model. The learning curve is a valuable tool for assessing how well a model is learning from the data and whether it is underfitting or overfitting.

What are the differences between supervised and unsupervised learning?

>supervised model has labeled dataset
>supervised model used for task like classification,predict numerical values
>In unsupervised model we don’t have the target variable
>unsupervised models often used to uncover hidden structures from the data for example anomaly

Classification+regression:

Decision tree model: it used for both regression+classification (DecisionTreeClassifier+DecisionTreeRegressor)

The decision tree algorithm based on the tree structure it takes the inputed data , than chooses the optimal root node based on the entropy metrics to classify the data based on it , and the classification stopes when the desired purity in the leaf nodes is reached or the max depth is reached, for both binary classification and multiclass classification tasks.

decision tree hypermaters:

depth,leafs,metric(gini,entropy)

Random forest model:

Random forest is built upon decision tree algorithm , it takes the data and split it into random subsets than for each data subset it applies the decision tree algorithm to it and the result is the average value of the each decision tree for the data subsets and in classification the result combined through majority vote .
n_estimators specify the number of trees.

how can you use the perfect number of n_estimators:
Cross-Validation: Use cross-validation to evaluate model performance for different values of `n_estimators`. By plotting the cross-validation error against different numbers of trees, you can identify the point where adding more trees results in diminishing returns in terms of accuracy improvement.

Majority vote:

- Decision Tree 1: Predicts "Apple"
- Decision Tree 2: Predicts "Banana"
- Decision Tree 3: Predicts "Apple"
- Decision Tree 4: Predicts "Apple"
- Decision Tree 5: Predicts "Apple"

In this case, "Apple" was predicted by 4 out of 5 trees, so the majority vote result is "Apple." Therefore, the final prediction for the input image is "Apple."

Gradient boost : regression+classification

Gradient boost begins with an initial prediction in regression task average value and in classification task a class distribution by then it creates a weak learner which is a decision tree with limited number of depth and multiply it with a learning rate and then the next weak learner made by taking the mistakes of the previous one which are the residuals . The process stops based on stopping criteria whether it is the number of epochs is reached or the overall loss is minimized.

hyperparameters:
learning rate,max_depth of the weak learner,number of epochs

Xgboost:

XGBoost builds upon the principles of gradient boosting but includes several enhancements and optimizations:
it includes regularization , efficiency to speed up the calculation , handels the missing data better (It can automatically decide how to split nodes with missing values, reducing the need for preprocessing.)

Adaboost:

In adaboost algorithms , it combines multiple weak learners which are stumps (decision tree with one level) and each stump is made by taking the mistakes of the previous stump into account.

example if we have to classify if a patient has a heart disease :
We have stumps that will classify the patient that he has heart disease and their total amount of say is superior to the amount of say of the stumps that classified the patient that he has not a heart disease .
hyperparameters:
n_estimators:numbers of weak learners you want to create
max_depth_of_n_estimators:control the max depth of the weak learners typically they are stumps

svm model:

The svm model used for classification tasks, face detection, and even charter reconginization
It separates the classes using n dimensional hyperplane based on a specific margin .
nearby data points called support vectors.
The margin is the distance between the hyperplane and the data point.
A larger margin is typically preferred because it often leads to better generalization.
the hyperplane can be controlled using the
c regularization (A smaller value of C creates a wider margin, but may lead to more misclassified data points. A larger value of C aims for a smaller margin, with fewer misclassifications.)
kernel type (linear, polynomial)

Kneighbors classifier:

The k_neighbors classifier is a supervised ML mode that takes the data point then assigns it to the class that has the k-nearest neighbors using distance metric.

hyperparameters:

k,distance metric

Logistic regression:

It uses the sigmoid function to model the relationship between categorical target variable and the other features in order to predict binary output.

Linear regression:

It uses a linear function to model the relationship between the target variable and the other inputs.

classification evaluation:

1.cost matrix : it is a way to assign different costs or penalties to different types of classification errors

define a cost matrix and multiply it by the confusion matrix

For example, the cost of predicting a patient who has no cancer is higher that predicting a patient that does not have cancer because it can cause the death of a person.

2.f1_score ,recall,precision,accuracy(classification report)

3.roc >for binary classification

roc: the roc graph is represented by the sensitivity(the ratio of the true positive) by the specificity (the ratio of the false negative)

auc: to compare the different roc curves

Segmentation:

unsupervised ml task , it's grouping similar data points in clusters based on distance metric.

elbow method: Given a range of k numbers, K-Means calculates the sum of squared distances between the data points and their centroids using the 'inertia' algorithm. When you plot the sum of squared distances (inertia) against the range of k, the optimal number of clusters is typically determined by identifying the 'elbow point' on the plot.

The elbow point represents a point,where increasing the number of clusters beyond this point does not result in a significant reduction in the sum of squared distances.

K-means:

K means starts by taking k random centroids data points and then calculates the distance between data points and the current centroids (cluster centers),After calculating the distances between data points and the centroids, the algorithm updates the centroids as the mean coordinates (center) of the data points in each cluster.The process ends when the data points stop changing clusters

DBSCAN:

DBSCAN groups together data points that are close to each other and separates areas with lower point density.

how you evaluate segmentation problem result:

Silhouette Score:

The score ranges from -1 to 1to evaluate the quality of clusters created by clustering algorithms, such as K-Means.

NLP

Transformers:neural network architecture : Attention is what you need

transformers encoder-decoder (translation)

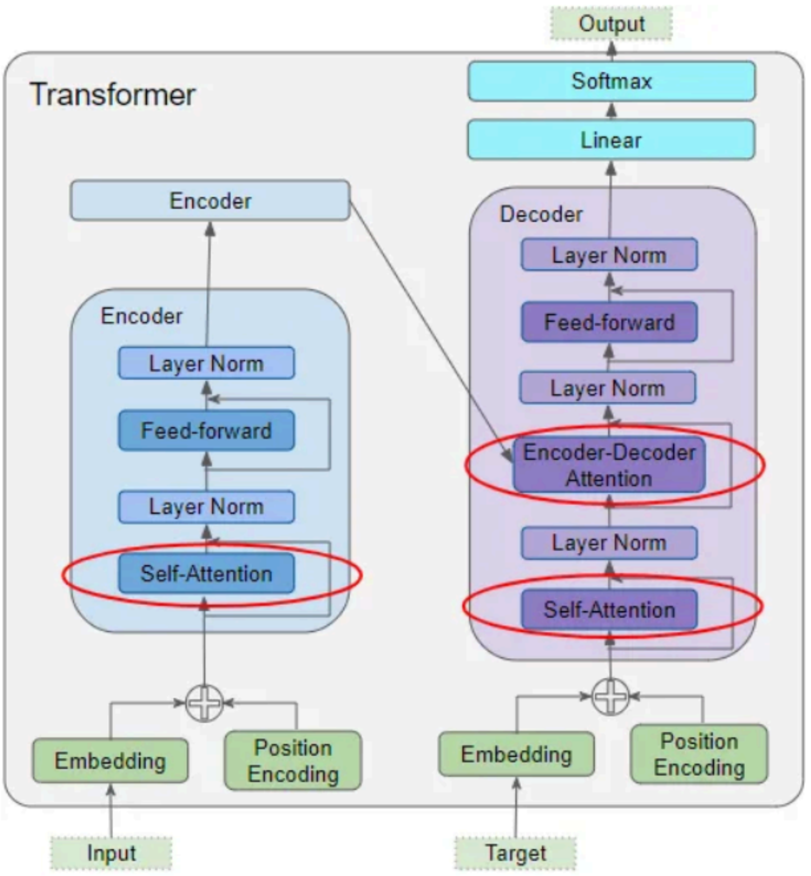
we are going to take the example of the sentence in english “let’s go” and we are looking to do translation in spanish ‘vamos’

The first step is to do the encoding:

1. Assign random vectors and weights to each token
2. Assign the positional argument values , which are the representation of the tokens positions in the sentence
3.Apply the self attention mechanism which calculates the similarity value between each token. This helps the model capture the relationships between words in the sentence.
4.residual connection which is the random vectors +positional arg+self attention result
5.Then layer normalization
6.feed forward layer to add complexity and so the models learn more from the patterns and capture more info
7.layer normalization

The second step is to do the decoding:

1. the first step begin with <sos> token as the first input of the decoder , encode it
2. pass the <sos> through encoder-decoder attention layer so it can connect with the econding layer to gain context the input words with the output
3. the output of the encoder-decoder attention is the first word we need to start with which is ‘let’
4. the word let’ then passed through fully interconnected layer and softmax function that generate the first word in spanish ‘vamos’
5. this process is repeated until we reach <eos>



(Image by Author)

self attention : where the magic happens

Self-attention is a mechanism that calculates how much each word in the sentence should pay attention to other words

The self attention mechanism is used to give to each tokens weights depending on their importance so the model knows what words he should pay attention to.
query,key,value
A single self-attention mechanism is called a head.

##transformer decoder only (used by gpt)

will take the example what is llm made .. and to complete with for
1.Initially begins with <sos>
2.encode the <sos> by pre trained word embedding
3Then assign positional encoding to the <sos>
5.the process stops when the decoder reachs <eos>

transformer encoder only

bert encoder(you can fine tune it for a task) , word2vec

word embedding existing tools and why we use pretrained models:

because of the cost , time saving , it requires smaller labeled data

fine tuning:

Fine-tuning involves customizing a pre-trained model for a specific task. Initially, you choose a related pre-trained model, like BERT for sentiment analysis. Instead of only replacing the output layer, you may modify the model's architecture, freeze lower layers, and fine-tune upper layers to adapt it. Then, you train the model on task-specific data, such as movie reviews and their sentiment labels. This approach combines the model's prior knowledge with specific task training to achieve better results.

transfer learning:

Fine-tuning involves customizing a pre-trained model for a specific task. Initially, you choose a related pre-trained model, like BERT for sentiment analysis. and you replace the output layer.

LLM:model based on the transformer architecture :

Large Language Models refer to a specific category of transformer-based models that have been trained on massive amounts of text data and have a vast number of parameters. Examples of LLMs include GPT-3, GPT-4, T5, and BERT (when fine-tuned on a large scale).

+ fine-tuning
+ rag systems
+ langchain
+ chuncking

#bag of words