**Neural Network**

1. **Fully Connected Layer**

The nodes in fully connected networks are commonly referred to as “neurons.” Here is an example with one hidden layer, one output layer and one input layer. This is the feed-forward neural network.

Diagram, bubble chart

Description automatically generated

To get the parameters through stochastic gradient descent:

* Initialization: Initialize all weights and bias to small random numbers.
* Forward pass: Pick a training example, x and feed example through the network to get the output y. And repeat this process to get the process until convergence.
* Back propagation and gradient descent to update parameters.

**Stochastic gradient descent:** Compute error on a single example at a time.

**Batch gradient descent:** Compute error on all examples.

▪ Feedforward all training data, compute the corresponding gradients.

▪ Take the average of the gradients of all the training examples.

▪ Use the mean gradients to update weights.

▪ repeat.

**Mini-batch gradient descent:** Compute error on small subset.

▪ Randomly select a “mini-batch” (i.e. subset of training examples).

▪ Calculate error on mini-batch, apply to update weights, and repeat.

**Activation function:** non-linear activation contributes to the non-linearity of the neural network,including Sigmoid, ReLU, Tanh functions.

1. **CNN**

Different with fully connected layer, CNN Connect each hidden unit to a patch of and share matrix of parameters across units. Here is an example of convolution neural network in 2D:

Diagram

Description automatically generated

**Zero padding:** to ensure the input and output have the same size.

**Application

Description automatically generated with medium confidence**

**Stride:** a parameter of the neural network's filter that modifies the amount of movement over the image or video. For example, if a neural network's stride is set to 1, the filter will move one pixel, or unit,  at a time. The size of the filter affects the encoded output volume, so stride is often set to a whole integer, rather than a fraction or decimal.

Table

Description automatically generated

**RGB:** red, green and blue, three layers.

**Pooling:** A pooling layer down samples its input.

A picture containing table

Description automatically generated

**Dropout Regularization:** A node cannot rely heavily on a particular node from previous layer.

Weights should be more spread.

Diagram, schematic

Description automatically generated

1. **RNN**

**Recurrent neural network** uses their internal state (memory) to process variable length sequences of inputs.RNN can deal with problems when the input length is too long or the input is of different size. The parameter for each input is all shared.

Diagram

Description automatically generated

Text, letter

Description automatically generated

For long sequences, RNN get into the vanishing gradient, so Long short-term memory (LSTM) is introduced. Gates are introduced to control the information flow.

Diagram

Description automatically generated

**Pytorch**

Pytorch will take care of the derivative computations. And we need to do following things to build neural network framework:

* Build the forward model (i.e., the computation graph).
* The training examples and training schedule (e.g., how the points are batched together).
* The optimization details (e.g., loss function, learning rate)

1 dataset

Graphical user interface

Description automatically generatedThe form of dataset in pytorch is tensor. For any dataset, it should be changed to tensor, otherwise, the code won’t go right. And dataset should be put in CPU or GPU. Remember that the model and dataset should on the same device (CPU or GPU), otherwise there might be some errors. Here are some data types.

If we want to transfer data from one device(CPU) to the other(GPU==cuda), here is an example:

data.to(cuda)

2 dataloader

Link: https://pytorch.org/docs/stable/data.html?highlight=dataloader#torch.utils.data.DataLoader

Dataloader combines a dataset and a sampler, and provides an iterable over the given dataset.

* Dataset:the dataset that we get from last step
* Batch\_size: the number of training examples in one forward/backward pass

Text, letter

Description automatically generated

3 model

Model layers are constructed by torch.nn module.

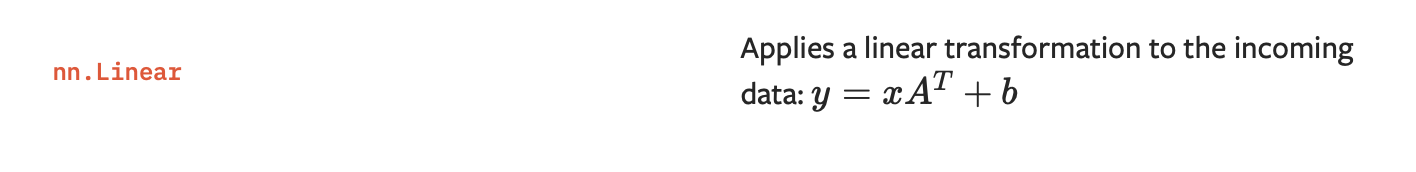
Link: <https://pytorch.org/docs/stable/nn.html>

Here are some commonly used layers.

Graphical user interface, application, Teams

Description automatically generatedGraphical user interface, application, Teams

Description automatically generatedGraphical user interface, application, Teams

Description automatically generated

Text, letter

Description automatically generatedText

Description automatically generated

4 training process:

Graphical user interface, text, application

Description automatically generated