پروژه 5 ام هوش مصنوعی NN -

سپهر آزردار

810199357

* **بخش سوم: طبقه بندی داده ها**

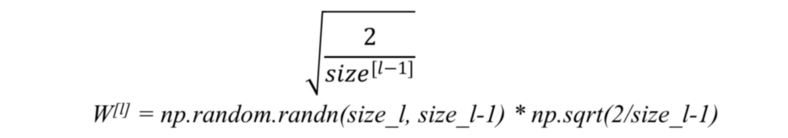
1. قسمت دوم وزن دهی شبکه
   1. اگر مقدار اولیه تمام وزنهای شبکه برابر صفر قرار بود و شبکه را آموزش میدادید، نتیجه آن چه بود؟

وقتی همه وزن ها یکی هستند برای تمام نورون ها ی یک لایه، input,output, derative یکسان است. پس ما در واقع همه چی متقارن هست و ما یک مدل خطی داریم.

hat it's a good idea to choose initial weights of a neural network from the range

(−1d√,1d√), where d is the number of inputs to a given neuron

If all the weights are initialized with 0, the derivative with respect to loss function is the same for every w in W[l], thus all weights have the same value in subsequent iterations. This makes hidden units symmetric and continues for all the n iterations i.e. setting weights to 0 does not make it better than a linear model.



1. Activation function
   1. دلیل اینکه Tanh و Sigmoid عملکرد مناسبی برای این دست شبکهها ندارند را بیان کنید

the tanh function was preferred over the sigmoid activation function as models that used it were easier to train and often had better predictive performance.

A general problem with both the sigmoid and tanh functions is that they saturate.

the functions are only really sensitive to changes around their mid-point of their input, such as 0.5 for sigmoid and 0.0 for tanh.

The amount of error decreases dramatically with each additional layer through which it is propagated . This is called the vanishing gradient problem and prevents deep (multi-layered) networks from learning effectively.( premature convergence,)

the error gradient can be unstable in deep neural networks and not only vanish, but also explode, where the gradient exponentially increases as it is propagated backward through the network. This is referred to as the “exploding gradient” problem.

an activation function is needed that looks and acts like a linear function, but is, in fact, a nonlinear function allowing complex relationships in the data to be learned.

* 1. برتری Relu Leaky نسبت به Relu چیست؟

large weight updates can mean that the summed input to the activation function is always negative, regardless of the input to the network.

a node with this problem will forever output an activation value of 0.0. This is referred to as a “dying ReLU“. the gradient is 0 whenever the unit is not active

The Leaky ReLU (LReLU or LReL) modifies the function to allow small negative values when the input is less than zero. The leaky rectifier allows for a small, non-zero gradient when the unit is saturated and not active; ELUs have negative values which pushes the mean of the activations closer to zero. Mean activations that are closer to zero enable faster learning as they bring the gradient closer to the natural gradient

1. Batch size
   1. علت استفاده از batch در فرایندآموزش چیست؟ مزایا و معایب size batch بسیار کوچک و بسیار بزرگ را شرح دهید

too large of a batch size will lead to poor generalization

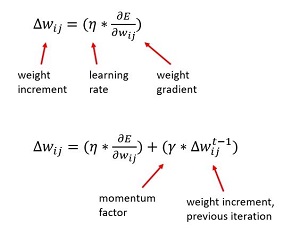
using a batch equal to the entire dataset guarantees convergence to the global optima of the objective function. But it is slower. But smaller batches have faster convergence to “good” solutions. smaller batch sizes allow the model to “start learning before having to see all the data.” the model is not guaranteed to converge to the global optima. It will bounce around the global optima, staying outside some ϵ-ball of the optima where ϵ depends on the ratio of the batch size to the dataset size.

starts at a small batch size, reaping the benefits of faster training dynamics, and steadily grows the batch size through training, also reaping the benefits of guaranteed convergence.

he reason for better generalization is vaguely attributed to the existence to “noise” in small batch size training.

* **بخش چهارم( استفاده از کتابخانه Tensorflow و رابط (Keras**
* تحقیق کنید که momentum چیست و چرا استفاده از آن در مرحله train مفید است؟

Momentum in neural networks is a variant of the stochastic gradient descent. It replaces the gradient with a momentum which is an aggregate of gradients.

Beside others, momentum is known to speed up learning and to help not getting stuck in local minima.

during training the update direction tends to resist change when momentum is added to the update scheme. When the neural net approaches a shallow local minimum it's like applying brakes but not sufficient to instantly affect the update direction and magnitude. Hence the neural nets trained this way will overshoot past smaller local minima points and only stop in a deeper global minimum

e value for the hyperparameter is defined in the range 0.0 to 1.0 and often has a value close to 1.0, such as 0.8, 0.9, or 0.99. A momentum of 0.0 is the same as gradient descent without momentum.

* آیا در همه مسائل نیاز به آن است که شبکه عصبی در چندین epoch تمرین نماید؟ دلیل این مسئله چیست؟

Each epoch is equal to one iteration over the whole dataset. Each epoch may include one or multiple batches that split the data set into separate groups. After iteration on the whole dataset, our Weights get updated a little. It always takes multiple iterations for Weights to converge. So we always need multiple iterations to converge by little updates.

* آیا همواره استفاده از تعداد epoch های بیشتر برای تمرین مفید است؟ اگر جواب شما مثبت است، دلیل خود را توضیح دهید و اگر جواب شما منفی است، راهحلهای مقابله با اتفاق نامطلوبی که رخ میدهد را بیان کنید.

No. if the number of epochs is too small, we may suffer from underfitting. Because the model doesn’t get enough time to find the underlying pattern. On the other hand, if the number is too high we may suffer from overfitting. Perform best on training data but not good on tests.

The solution is to run the model as long as possible and halt the process whenever the overfitting is going to happen.

**MSE**

penalizes errors in an incompatible way for discrete distributions. And doesn’t treat differently FN, FP.

When you derive the cost function from the aspect of probability and distribution, you can observe that MSE happens when you assume the error follows Normal Distribution and cross-entropy when you assume binomial distribution. It means that implicitly when you use MSE, you are doing regression (estimation) and when you use CE, you are doing classification.

Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. It is preferred for classification, while mean squared error (MSE) is one of the best choices for regression