1.3

O método de gradient descente é usado para minimizar o risco empírico para uma determinada função de custo(POR ESSAS EQS), isto é feito através da adaptacção dos segundo as seguintes equações para o modo de treino em *batch* e usando momentum :

Onde corresponde ao peso do ramo com nó inical i e final j, (i=0 para DC)

Para calular o é necessário primeiro fazer na rede uma propagação da rede para a frente de forma a obtermos os e posteriormente uma propagação para trás obtendo os

(ESTA CURTA DEMAIS)

StepSize = 0.9

Momentum term = 0.5

(Qual a diferenca entre epoch e tempo? Nao sao lineares?)

Mediana dos epoch =235

(E’ preciso justificar alguma coisa)

1.4)

A expressão permanece igual à representada na alínea 1.3 com uma pequena adição, o é agora também modificado a cada iteração sendo que:

Recalcular epochs <-params do prof? 87

Os valores obtidos são 0.9 para o learning rate e 0.7 para o momentum term

Para o mesmo momentum term () como este algoritmo é mais resistente a variações dos parâmetros, a variação do número de epochs é bastante baixo, da ordem das unidades,

No caso em que não existe passo adaptativo ocorrem grandes alteração o número de epochs ao variar minimamente os parametros. Como neste caso estamos a usar passo adaptativo, o valor inicial de tem pouca importância

Matriz confusao

Os percentagem de vezes que a rede acerta em determinado padrão à entrada é cerca de 73% o que é satisfatório, Sendo que se obtém diferentes percentagens para dígitos diferentes.A causa destes erros advém das similaridades dos dígitos, pode por vezes confucar o dígito 8 com o 9, o 1 com o 7. De notar que o dígito 8 costuma ter taxa de erro relativamente alta para a maioria de testes, ronda os 60% de classificações erradas. Sendo que a accuracy varia devido ao pesos inicias serem diferentes entre diferentes treinos. Estavamos à espera de uma test set accuraccy de cerca de 80%.

Training error <- Corresponde ao MSE final? Probably not= recalcular 0.05492

Test accuraccy 73.3%

Ambos os parâmetros são relevantes no entanto é mais fácil averiguar a qualidade da rede através do test set accuracy, pois nos é dito logo qual a percentagem de sucesso. Além disso pode ocorrer overfitting, se fizer quisermos aumentar a accurancy pendindo um MSE mais baixo poderá causar overfitting, que poderá causar que a rede seja menos flexível e porudza um test set accuracy mais baixo.

//nao é verificável no MSE mas que entende-se que existe ao observar a test accuracy

Os pesos são escolhidos de forma a obter um MSE mínimo para o grupo de validação, isto é feito para cada epoch, qual o erro de valição qual o menor. O objectivo é escolher o grupo de validação como um grupo de dados “do mundo real” que erá aplicado à rede para que os pesos escolhidos se apliquem da melhor forma ao modelo criado, com este passo é possível prevenir o overfitting dos modelos.

No plot representados oberva-se que no inicio o MSE que todos os conjuntos de treinos estão a convergir para o goal MSE definido por nós. Podemos verifica que o MSE do validation set de cada época que nos diz qual o pesos que devemos utilizar para posterior utilização da rede, podemos ver que para as ultimas época dos training set os MSE correspondentes do test e validation estão a aumentar, o que é indicio de overfitting da rede.

Como se pode observar no gráfico à esquerda a rede faz um bom fitting do conjunto de treino, sendo que apresenta overfitting, não se assemelhando muito à função que cos(5x)

Sim temos indícios de overfitting, pois para os pesos após o treino da rede vão gerar um valor do MSE muito grande para o conjunto de treino, ou seja a função gerada vai-se encontrar afastada dos ponto de teste e muito próxima(é imposto um valor muito baixo como condição de paragem) dos valores de treino.

Sim existe overfitting pois a função tenta passar por todos os pontos de treino, causando que existam picos que se afastam muito da função inicial que se queria obter.