PLAGIARISM SCAN REPORT

Words	780	Date	April 22,2021
Characters	4938	Excluded URL	

0% Plagiarism	100% Unique	0 Plagiarized Sentences	46 Unique Sentences

Content Checked For Plagiarism

A.1 ARTIFICIAL NEURAL NETWORKS
Figure A.1 shows a schematic of the simplest multi-layer perceptron network, i.e. a feed-forward neural network. The network consists of an input layer, hidden layers and output layer. Define a`
i as the ith neuron of the `th layer and a`+1
j as the
jth neuron of the (`+1)th layer and w`
i j;b`
j is the weight and bias connecting the two neurons, then the output of the `th layer is given by:
The loss is calculated using a loss function such as cross entropy function especially for binary classification.
where y0 2 f0;1g;y 2 (0;1). This objective is to minimize this cross entropy over a batch of all training data. This is done using gradient descent where the parameters viz. weights and biases are updated to reduce the overall cross entropy loss.
A.2 CONVOLUTIONAL NEURAL NETWORKS
A convolution is a linear operation that can be viewed as a multiplication or dot product of matrices. The input is a tensor of shape height x width x channels and the convolution operation abstracts the image to a feature map (also called a kernel) of shape kernelsize x kernelsize x kernelchannels. The layers can be computed by: where ` is the layer, g is the activation function ReLU, Fj is the receptive field, k is the convolutional kernel and b is the bias.
A.3 RESIDUAL NETWORKS
Figure A.2 shows a building block of a residual network. The residual blocks can be expressed mathematically as follows. Let h(a) be an underlying mapping that is to be fit by a set of layers, where a is the input to these layers. If we hypothesize that multiple non-linear transformations by these layers can approximate the layer functions, then one can also hypothesize that a similar approximation can be made for the residual functions, i.e. h(a)□a, which have the same input and output dimensions. So instead of letting the underlying mapping be h(a) we approximate a residual function F(a) = h(a)□a. Thus, the actual function becomes h(a) = F(a)+a. We can achieve this approximation in a feed forward neural network using a series of skip connections that perform identity mapping and jump over a few layers. Adding the outputs of these two connections gives the final output layer. Thus, the residual unit can be defined as,
where a`+1 and a` represent the input and output for the `th layer and F is the residual function. The h denotes the operation in the identity mapping. The output of the layer is generally processed using an activation function such as ReLU. Let f be the ReLU activation and replacing F with the definition of feed forward activation, the

residual block thus can be defined as,
For the sake of simplicity, we consider no operation being performed in the identity
mapping. Thus, equation A.4 and equation A.5 become
A.4 GENERATIVE ADVERSARIAL NETWORKS
A generative network, G, is supposed to learn the underlying distribution of a latent
space, Y. Instead of visually assessing the quality of network outputs and judge how
we can adapt the network to produce convincing results, we incorporate automatic
tweaking during training by introducing a discriminative network D. The network D
takes in both the fabricated outputs generated by G and real inputs from the underlying
distribution Y. The network produces a probability of the image belonging to
the real or fabricated space.
Let x 2 X be a low resolution/grayscale image and y 2 Y be it's underlying
distribution from the latent space Y. Generator G takes in input x and produces an
output ˆ y. We define the mapping x! ˆ y in the following manner:
The discriminative network D is fed the fabricated mapping x ! ˆ y and the
underlying distribution of x i.e. y 2 Y. The network D then produces a result that is
a probability distribution of the input space indicating the class of the image that it thinks the input belongs to. We define this as:
where p 2 (0;1) is the probability that the image is fabricated or real. Both generator and discriminator are contitioned on the input x in conditional GAN. Let
the generator be parameterized by qg and the discriminator be parameterized by qd.
The minimax objective function can be defined as:
Where, Gqg is the output of the generator and Dqd is the output of the discriminator.
We're currently not introducing any noise in our generator to keep things
simple for the time being. Also, we consider L1 difference between input x and output
y in generator. On each iteration, the discriminator would maximize qd according
to the above expression and generator would minimize qg in the following way:

Sources	Similarity