ROSHAN POUDEL - HW#3 Problem 4

Supervised learning is the most common form of machine learning. It involves training a machine to classify input data into specific categories by providing it with a large dataset of labeled examples. During training, the machine produces an output in the form of a vector of scores for each category, and an objective function measures the error between the output scores and the desired pattern of scores. The machine modifies its internal adjustable parameters, known as weights, to reduce this error. This is done by computing a gradient vector that indicates by what amount the error would increase or decrease if each weight were increased by a tiny amount, and adjusting the weight vector in the opposite direction to the gradient vector.

The objective function can be seen as a hilly landscape in high-dimensional space, with the negative gradient vector indicating the direction of the steepest descent towards a minimum, where the output error is low. We often use a procedure called stochastic gradient descent, which involves computing the outputs and errors for a few examples, computing the average gradient, and adjusting the weights accordingly. This process is repeated for many small sets of examples from the training set until we reach the global minimum of the loss function. After training, the system's performance is measured on a test set to test its generalization ability.

This article explains the backpropagation procedure in deep learning, which is a practical application of the chain rule for derivatives. The key idea is to compute the derivative of the objective with respect to the input of a module by working backward from the gradient with respect to the output of that module. This process can be repeated to propagate gradients through all modules, starting from the output and ending at the input. Once the gradients have been computed, the weights of each module can be updated. The article also explains feedforward neural network architectures, which use hidden layers to distort the input in a non-linear way so that categories become linearly separable by the output layer. The ReLU non-linear function is the most popular choice for the units in these architectures, as it learns much faster in networks with many layers.

Convolutional neural networks (ConvNets) are designed to process data in the form of multiple arrays, such as images or audio spectrograms. They utilize local connections, shared weights, pooling, and many layers to detect local features and merge semantically similar ones. The architecture consists of convolutional and pooling layers organized in feature maps, with units connected to local patches in the previous layer through a set of weights called a filter bank. The pooling layer reduces the dimension of the representation and creates invariance to small shifts and distortions. ConvNets are inspired by the classic notions of simple and complex cells in visual neuroscience and are reminiscent of the visual cortex ventral pathway. There have been numerous applications of ConvNets since the early 1990s, including speech and document recognition, optical character and handwriting recognition, and object detection and recognition.

Backpropagation has led to the development of RNNs, which are effective in handling sequential tasks like speech or text processing with varying input lengths. The RNNs have a memory element called "hidden state," which serves as an internal representation of the sequence by keeping track of previous time step information. However, training RNNs has been difficult due to the issue of vanishing or exploding gradients. Advanced RNNs like LSTM networks have been developed to overcome this challenge. These networks can selectively remember or forget information over extended periods, allowing for efficient handling of long-term dependencies.