TRAINING DEEP NEURAL-NETWORKS BASED ON UNRELIABLE LABELS

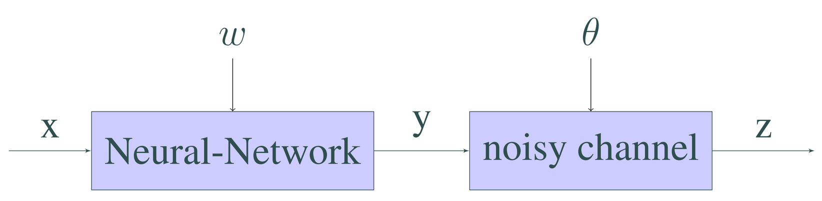
Alan Joseph Bekker   Jacob Goldberger

Faculty of Engineering, Bar-Ilan University, Israel

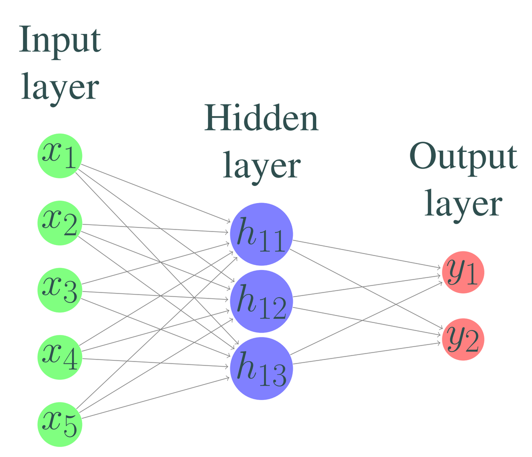
Main Objectives

- Improving the performance of a DNN while learning on data with noisy labels.
- Estimating the noise parameters.
- Showing the relevance of our model even when the labels are assumed to be noise free.

Model



A diagram of the model, the true label y is hidden and we observe a noisy version of it z.



A diagram of artificial neural network.

- Let $h = h(x)$ be the non-linear function applied on an input x. The soft-max output layer is:

$$p(y = i|x; w) = \frac{\exp(u_i^\top h)}{\sum_{j=1}^{k} \exp(u_j^\top h)}$$

  such that $u_1, ..., u_k$ are the soft-max parameters which are sub-set of the entire network parameter set w.
- The noisy-channel parameter is:

$$\theta(i, j) = p(z = j|y = i)$$

- The probability of observing a noisy label z given the feature vector x is:

$$p(z = j|x; w, \theta) = \sum_{i=1}^{k} p(z = j|y = i; \theta)p(y = i|x; w)$$

In the training phase we are given n feature vectors $x_1, ..., x_n$ with corresponding unreliable labels $z_1, ..., z_n$ which are viewed as noisy versions of the correct hidden labels $y_1, ..., y_n$. The log-likelihood of the model parameters is:

$$L(w, \theta) = \sum_{t=1}^{n} \log(\sum_{i=1}^{k} p(z_t|y_t = i; \theta)p(y_t = i|x_t; w))$$

Noisy-labels Neural-Network (NLNN) Algorithm

Input: Data-points $x_1, ..., x_n \in R^d$ with corresponding noisy labels $z_1, ..., z_n \in \{1, ..., k\}$.
Output: Neural-network parameters w and noise parameters θ.

The EM Algorithm iterates between the two steps:

E-step: Estimate true labels based on the current parameter values:

$$c_{ti} = p(y_t = i|x_t, z_t; w_0, \theta_0) = \frac{\theta_0(i, z_t) \exp(u_{i0}^\top h_0(x_t))}{\sum_j \theta_0(j, z_t) \exp(u_{j0}^\top h_0(x_t))}$$

M-step: Update the noise parameter θ:

$$\theta(i, j) = \frac{\sum_t c_{ti} 1_{\{z_t = j\}}}{\sum_t c_{ti}}$$

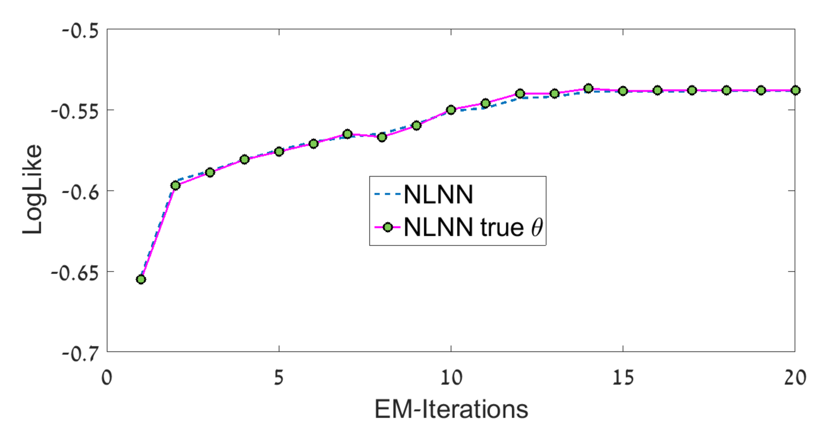
and train a NN to find w that maximizes the following function:

$$S(w) = \sum_{t=1}^{n} \sum_{i=1}^{k} c_{ti} \log p(y_t = i|x_t; w)$$

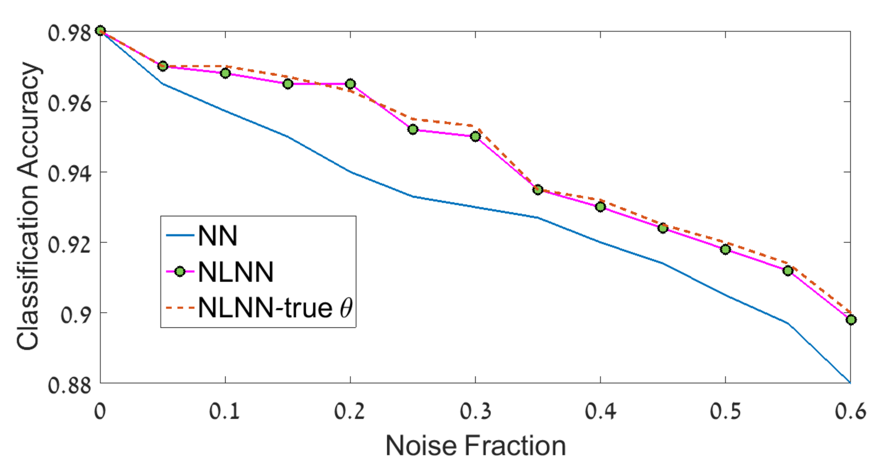
Back-propagation:

$$\frac{\partial S}{\partial u_i} = \sum_{t=1}^{n} (p(y_t = i|x_t, z_t; w_0, \theta_0) - p(y_t = i|x_t; w))h(x_t)$$

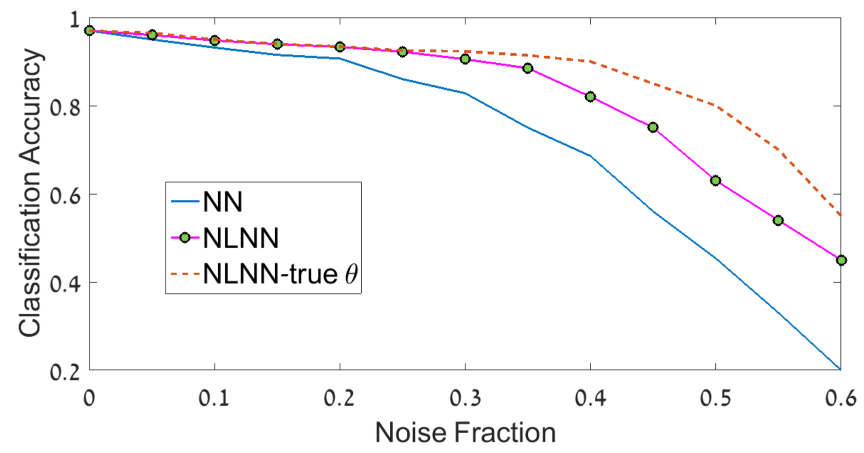
Results - MNIST



The model likelihood as a function of the EM iterations (purple), against a model where the true theta is known (blue).

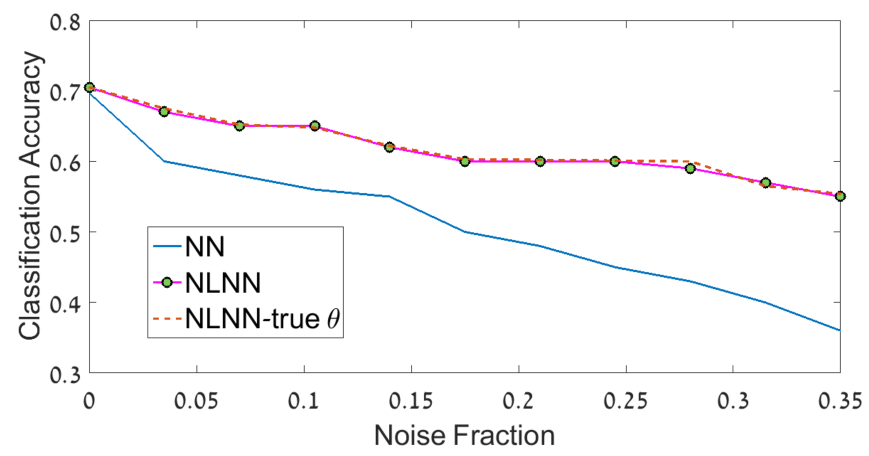


MNIST test data classification accuracy as a function of fraction of noisy labels with uniform noise.

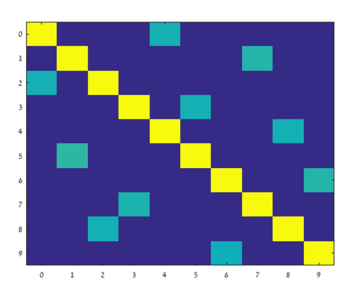


MNIST test data classification accuracy as a function of fraction of noisy labels with permutation type noise.

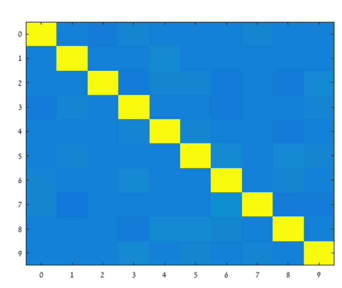
Results - TIMIT



Phoneme classification results as a function of the noise ratio on TIMIT data.



θ with permutation type noise.



θ with uniform noise.

Conclusions

- ✓ NLNN outperforms a regular NN for every noise fraction.
- ✓ NLNN correctly estimates the noise parameters.
- ✓ The algorithm can be easily incorporated into existing deep learning implementations.
- ✓ Our results encourage collecting more data at a cheaper price, since mistaken data labels can be less harmful to performance.
- ✓ Future directions: Generalize our learning scheme to cases where both the features and the labels are noisy.