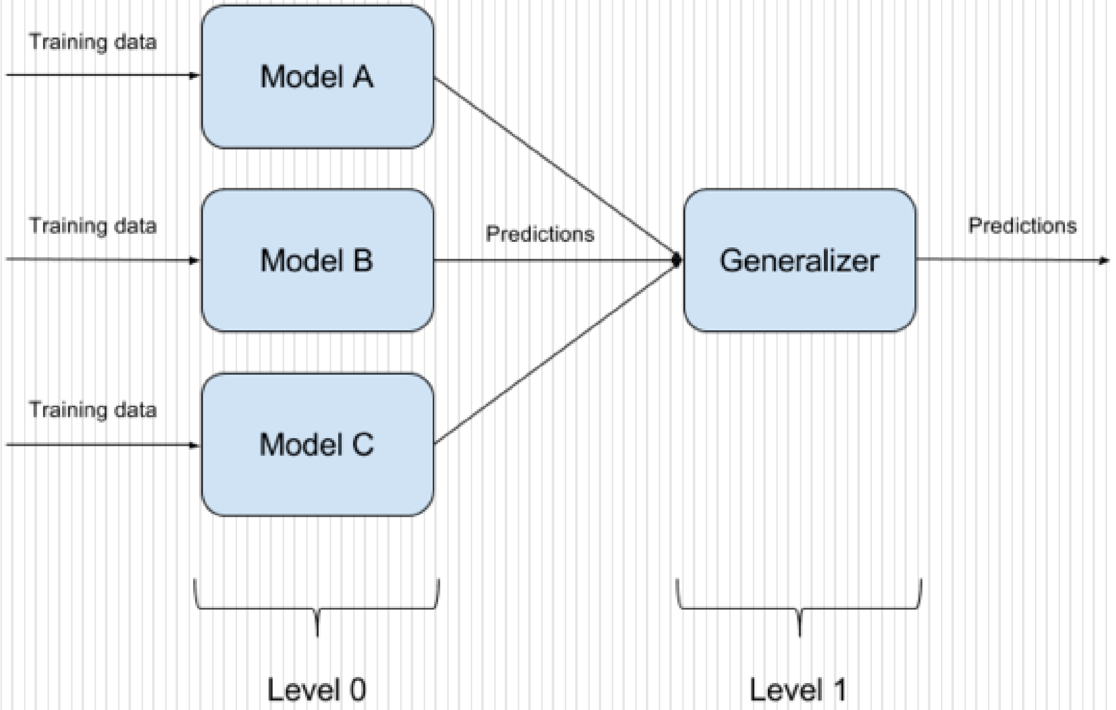
Amended Cross-Entropy cost: an approach for encouraging diversity in classification ensemble (Brief Announcement)

Ron Shoham and Haim Permuter

CSCML June 2019

Motivation

- Ensemble of models is a fundamental technique
- Diversity between the predictions is necessary
- For Regression - Negative Correlation Learning (NCL)
- For Classification ?



Why cross-entropy?

- Sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$
- MSE $\quad e_i = 0.5(\sigma(x) - y)^2$
- Gradient $\frac{\partial e_i}{\partial x} = (\sigma(x) - y)\sigma(x)\big(1 - \sigma(x)\big)$
- Saturation problem

We wish to get

- $\frac{\partial e_i}{\partial x} = (\sigma(x) - y)$
- $\frac{\partial e_i}{\partial \sigma(x)} = \frac{(\sigma(x)-y)}{\sigma(x)(1-\sigma(x)}$
- $e_i = \int \frac{\partial e_i}{\partial f_i} df_i, \quad f_i = \sigma(x)$

$$= -y\log(f_i) - (1-y)\log(1-f_i) + C$$

$$= H(y, f_i) + C$$

Negative Correlation Learning
Gavin Brown et al.

$$f_{ens} = \frac{1}{M}\sum f_i$$

$$e_i = \frac{1}{2}(f_i - t)^2 + \gamma(\mathrm{f_i} - \mathrm{f_{ens}})\sum_{j \neq i}(f_j - f_{ens}).$$

- Gradient analysis

$$\frac{\partial e_i}{\partial f_i} = (f_i - t) \; - \gamma[2\left(1 - \frac{1}{M}\right)(f_i - f_{ens})]$$

$$= (f_i - t) - \lambda(\mathrm{f_i} - \mathrm{f_{ens}})$$

$$= (1 - \lambda)(f_i - t) + \lambda(f_{ens} - t).$$

Amended cross-entropy

$$\frac{\partial e_i}{\partial z_i} = (1-\lambda)(f_i - y) + \lambda(f_{ens} - y)$$

$$\frac{\partial e_i}{\partial f_i} = \frac{(1-\lambda)(f_i - y) + \lambda(f_{ens} - y)}{f_i(1-f_i)}$$

$$= \frac{f_i - y}{f_i(1-f_i)} - \frac{\lambda}{M}\sum_{j \neq i}\frac{f_i - f_j}{f_i(1-f_i)}$$

$$e_i = \int \frac{\partial e_i}{\partial f_i} df_i$$

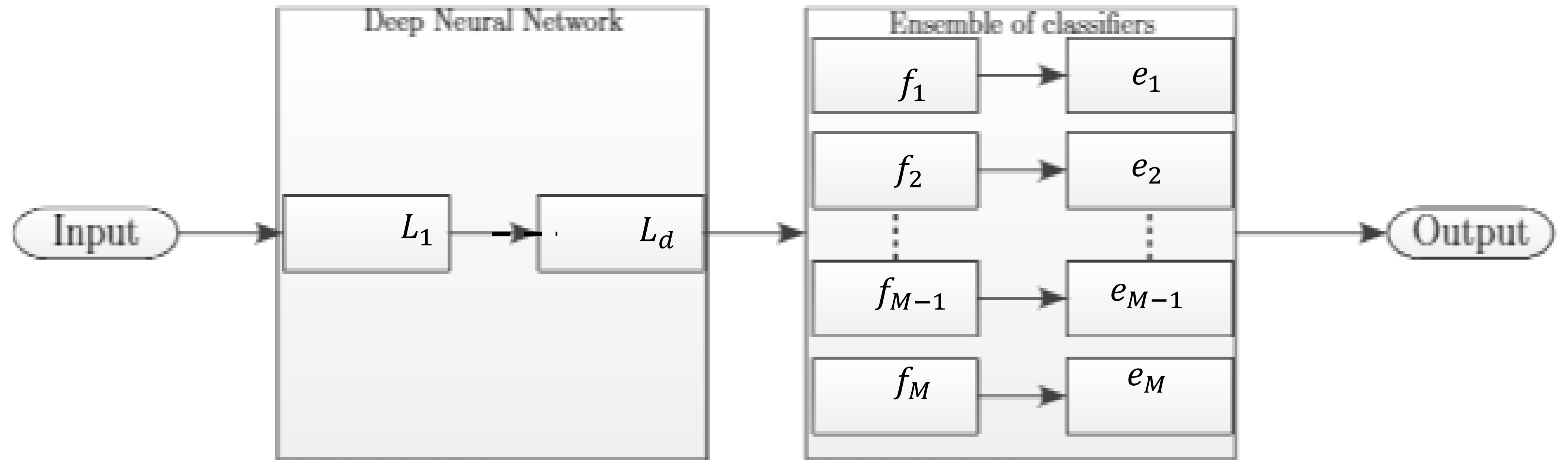
$$= -ylog(f_i) - (1-y)\log(1-f_i)$$

$$- \frac{\lambda}{M}\sum_{j \neq i}\{-f_j \log(f_i) - \left(1 - f_j\right)\log(1-f_i)\}$$

$$= H(y, f_i) - \frac{\lambda}{M} H(f_j, f_i)$$

Usage – Stacked Diversified Mixture of Classifiers

- Problem: Training an ensemble results in increasing the model.
- Solution: stacking a mixture of classifiers at the final layer of a Deep Neural Network



Results

- Database – Cifar 10
- Architecture – ResNet 110
- Number of parameters:
  - Original: 1731002 parameters
  - ResNet110 + SDMC (M=10): 1736852 parameters (+0.34%)

	$M=1$	$M=10$ $\lambda=0$	$M=10$ $\lambda=0.001$	$M=10$ $\lambda=0.01$	$M=10$ $\lambda=0.05$	$M=10$ $\lambda=0.1$	$M=10$ $\lambda=0.3$	$M=10$ $\lambda=0.5$
error(%)	6.43	6.2	6.14	6.12	**5.98**	6.09	6.13	6.31
CE	0.3056	0.3102	0.3041	0.3048	0.2968	**0.2918**	0.3137	0.4957

- Optimal lambda reduced error by ~7%