Deep Self-Taught Learning for Handwritten Character Recognition

Université de Montréal

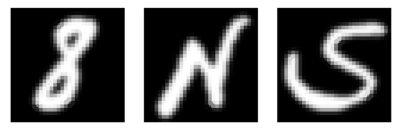
Yoshua Bengio, Frédéric Bastien, Arnaud Bergeron, Nicolas Boulanger-Lewandowski, Thomas Breuel, Youssouf Chherawala, Moustapha Cisse, Myriam Côté, Dumitru Erhan, Jeremy Eustache, Xavier Glorot, Xavier Muller, Sylvain Pannetier Lebeuf, Razvan Pascanu, Salah Rifai, François Savard, Guillaume Sicard

Experiments performed for the IFT6266 course
Winter 2010
given by Yoshua Bengio

Self-taught learning

Exploit out-of-distribution examples to improve results

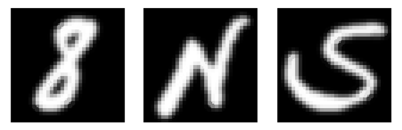
Pretrain, finetune on out-of-distribution



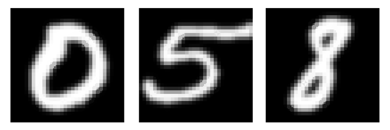
Test on these

Multi-task learning

Multiple tasks sharing parts of training, sharing parts of a model

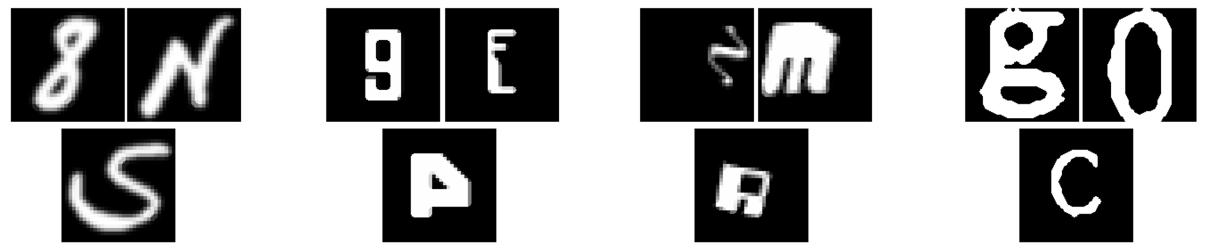


Pretrain, finetune on all classes



Test conditioning on digits only

Image sources



NIST (40%)
800k handwritten glyphs

Fonts (10%)
~3000 freely available fonts

CAPTCHAS (25%)
Fonts + transformations to approximate handwriting

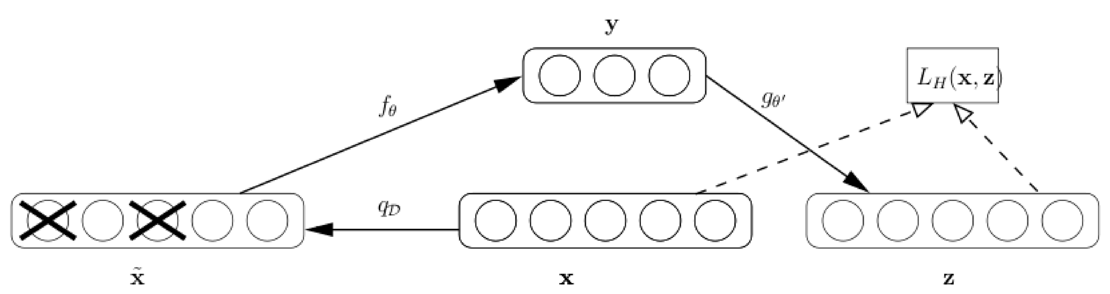
OCR (25%)
2 million scanned printed glyphs

Datasets used for training, testing

- All datasets are made of **32x32 grayscale images**
- **NIST (original dataset)**: NIST Special Database 19, 814,255 digits and upper/lowercase characters
- **NISTP**: images from **all 4 sources above** (with proportions given), with **transformations up to "local elastic distortions"**, complexity parameter of 0.7
  - Goal is to produce examples close to the ones found in NIST
- **P07**: images from all 4 sources (with proportions given), with **all transformations** and a complexity of 0.7.

Autoencoders

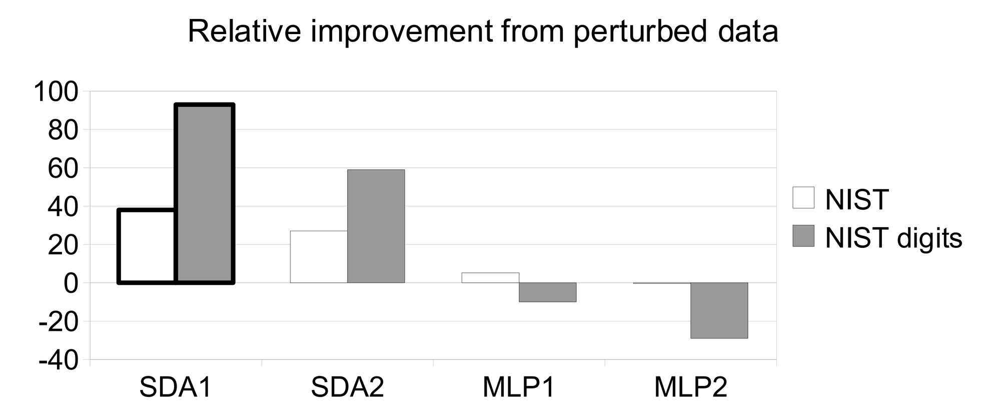
Show the network a corrupted input ($\tilde{x}$, corrupted from x) and train the parameters trying to reconstruct the original, uncorrupted version in z. The goal is to extract meaningful representations in y.



Hypotheses and conclusions

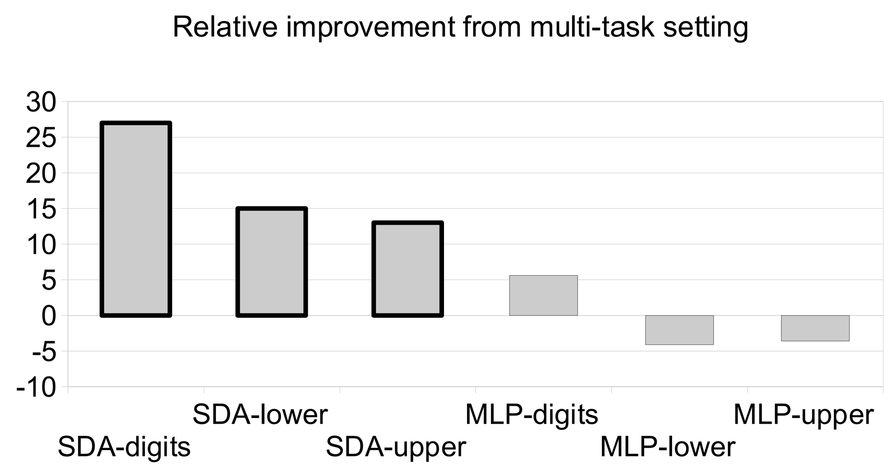
- **Deep learners benefit more** from out-of-distribution examples than shallow learners
  - Advantage preserved even as number of labeled examples available increases
  - Hypothesized explanation: lower layers learn low-level features which can be reused across tasks and for various distributions modeled in upper layers
- Good results of deep architectures on small datasets such as MNIST can scale to much larger ones
- Beats state of the art on NIST digits and reach human performance on clean 62 classes.

Relative improvement with self-taught and multi-task learning



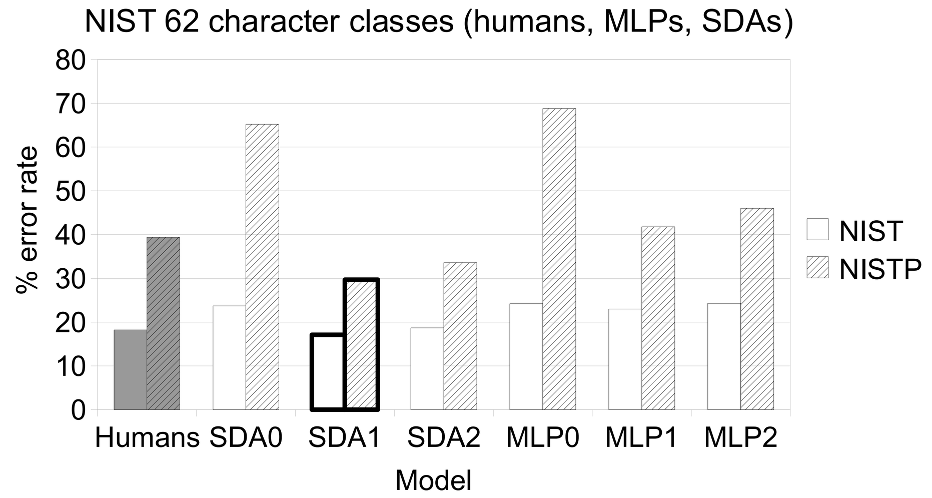
- For SDA[012], we compare pretraining and finetuning on P07 or NISTP, then testing on NIST, with training & test both done on NIST
- For MLP[012], we compare training for classification on P07 or NISTP, then testing on NIST, with training & test both on NIST

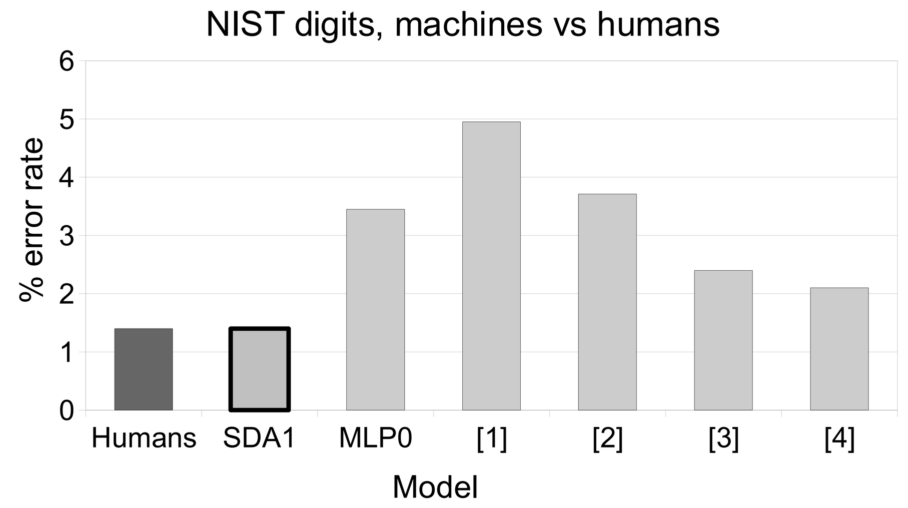
MLP0/SDA0 use NIST, MLP1/SDA1 use P07, MLP2/SDA2 use NISTP



- SDA-* here are pretrained and finetuned on all classes of NIST and tested knowing the class may only be a digit etc.
- MLP-* here are trained for classification on all classes, then tested knowing the class may only be a digit etc.
- The performance are compared to training on only digits, instead of all classes.

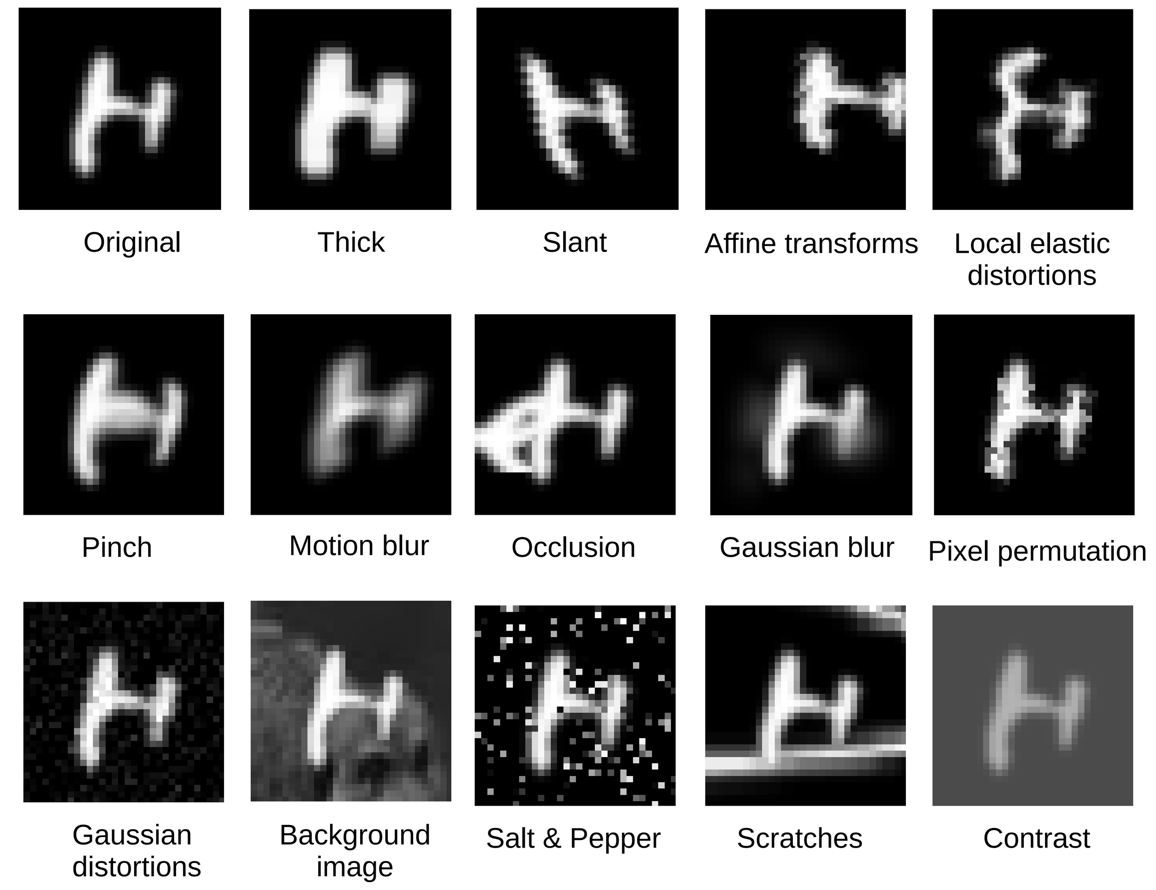
Absolute classification performance on NIST





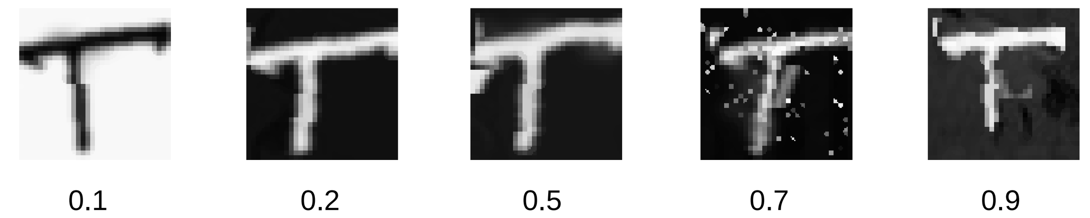
[1] Granger et al., 2007 (used ART) [2] Pérez-Cortes et al., 2000 (nearest neighbor) [3] Oliveira et al., 2002b (MLP) [4] Milgram et al., 2005 (SVM)

Transformations



Complexity parameter

Each transformation is controlled by a complexity from 0 to 1.



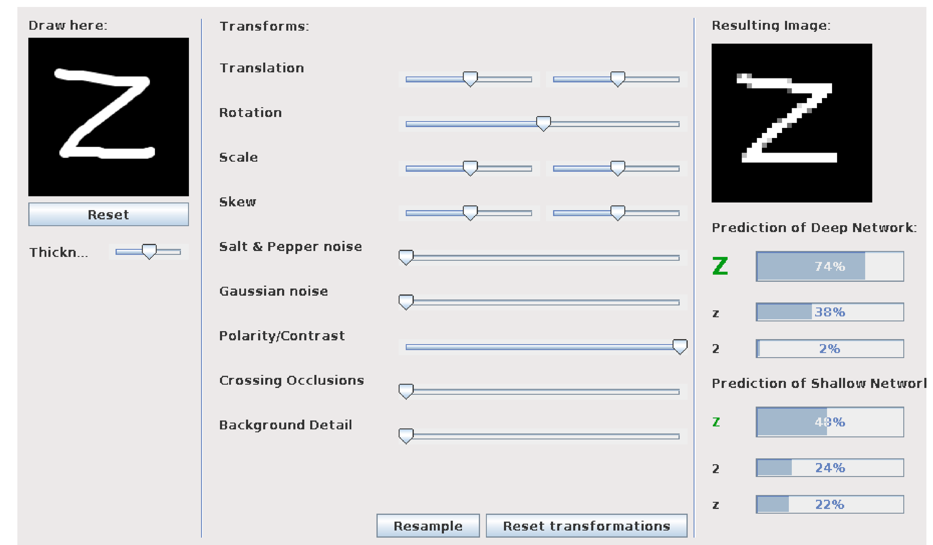
When transforming, each filter samples its own complexity up to pipeline maximum (for example sample over 0-0.7 for P07).

Human performance evaluation

- **Amazon Mechanical Turk**
  - Pay a small fee each time a very simple task is performed
- 2500 glyphs were classified by 80 persons
- Each glyph classified 3 times for error evaluation
- **Humans make mistakes, too: 18%!**
  - This is because **many glyphs look the same**: 1 or l; C or c. Z or z. etc.

Online Java applet demo

Draw a character, see instant classification. Runs client-side.



http://deep.host22.com