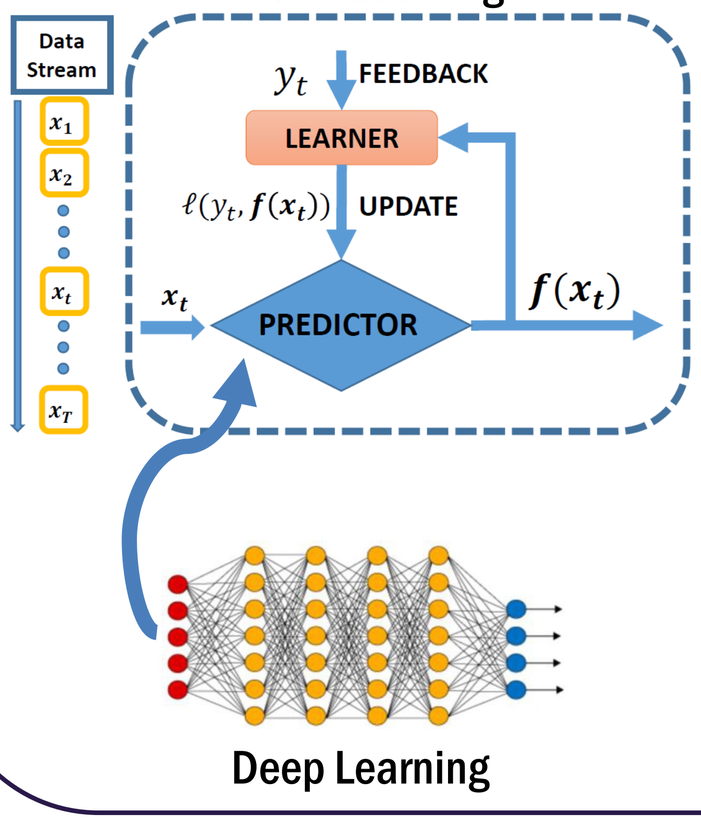
Online Deep Learning: Learning Deep Neural Networks on the Fly

Doyen Sahoo, Quang Pham, Jing Lu, Steven C. H. Hoi

Introduction and Motivation

Online Learning



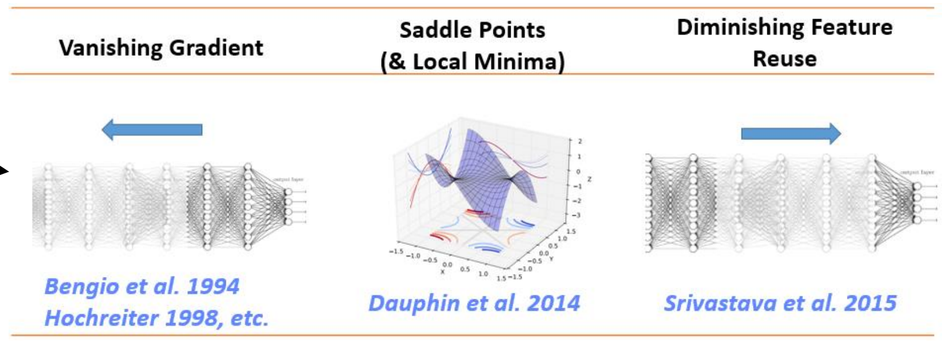
Can we use Deep Neural Networks to learn from data streams in an online setting?

Two major challenges

Convergence Issues

While using a Deep Network Online

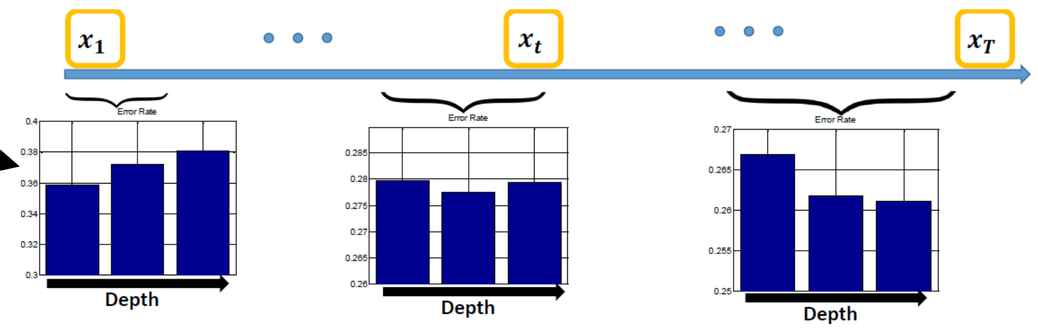
- We face Vanishing Gradient, Saddle Points and Diminishing Feature Reuse
- Batch Training can (with difficulty) overcome these issues by iterating through data multiple times
- Online Learning goes through the data only once, and is evaluated on its online performance



Model Selection

How do we choose depth prior to training?

- Shallow networks give better performance initially
- Deep Networks do better at a later stage
- Based on amount of data different network depth may give best overall performance
- Validation in online setting (particularly for depth selection) is non-trivial

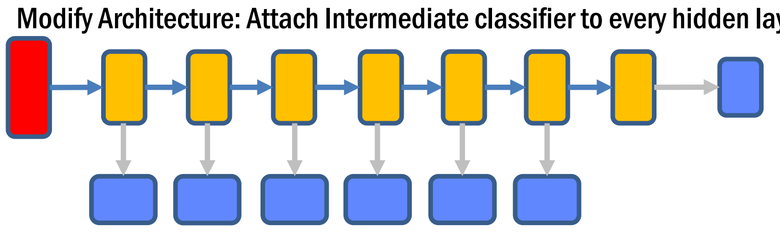


Online Deep Learning: Hedge Backpropagation

Shallow to Deep principle

- Start shallow → Fast Convergence
- Become Deeper → Deep Representation

Modify Architecture: Attach Intermediate classifier to every hidden layer



Feedforward

$$\begin{aligned}\mathbf{F}(\mathbf{x}) &= \sum_{l=0}^{L} \alpha^{(l)} \mathbf{f}^{(l)} \quad \text{where} \\ \mathbf{f}^{(l)} &= \text{softmax}(\mathbf{h}^{(l)}\Theta^{(l)}),\ \forall l = 0, \ldots, L \\ \mathbf{h}^{(l)} &= \sigma(W^{(l)}\mathbf{h}^{(l-1)}),\ \forall l = 1, \ldots, L \\ \mathbf{h}^{(0)} &= \mathbf{x}\end{aligned}$$

The feedforward function is modified from traditional DNNs, where the final output is a weighted combination of the outputs of all the intermediate classifiers. A dynamic Objective function is used for training

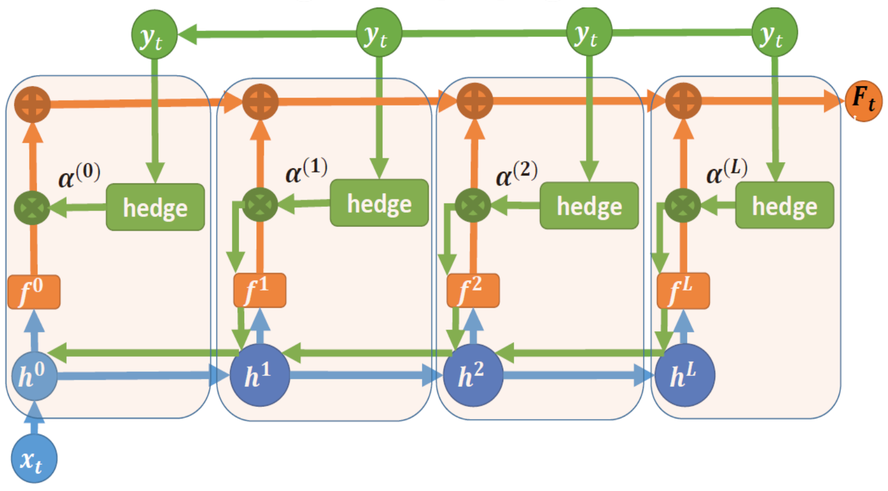
A Dynamic Objective Function

$$\mathcal{L}(\mathbf{F}(\mathbf{x}), y) = \textstyle\sum_{l=0}^{L} \alpha^{(l)} \mathcal{L}(\mathbf{f}^{(l)}(\mathbf{x}), y)$$

3 Main Updates

Loss / Classifier Weight update (Hedge)	$\alpha_{t+1}^{(l)} \leftarrow \alpha_t^{(l)} \beta^{\mathcal{L}(\mathbf{f}^{(l)}(\mathbf{x}), y)}$
Classifier Update	$\Theta_{t+1}^{(l)} \leftarrow \Theta_t^{(l)} - \eta \nabla_{\Theta_t^{(l)}} \mathcal{L}(\mathbf{F}(\mathbf{x}_t, y_t))$ $= \Theta_t^{(l)} - \eta \alpha^{(l)} \nabla_{\Theta_t^{(l)}} \mathcal{L}(\mathbf{f}^{(l)}, y_t)$
DNN Update	$W_{t+1}^{(l)} \leftarrow W_t^{(l)} - \eta \sum_{j=l}^{L} \alpha^{(j)} \nabla_{W^{(l)}} \mathcal{L}(\mathbf{f}^{(j)}, y_t)$

Hedge Backpropagation

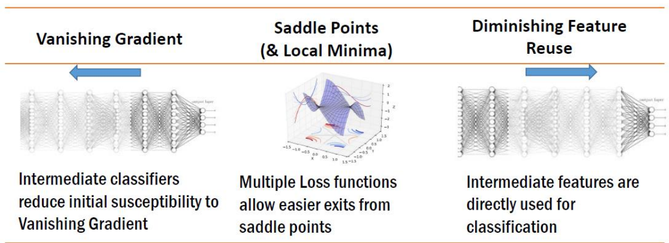


Online Deep Learning framework using Hedge backpropagation (HBP). Blue lines represent feedforward flow for computing hidden layer features. Orange lines indicate softmax output followed by the hedging combination at prediction time. Green lines indicate the online updating flows with the hedge backpropagation approach. In every round, due to the updates to α by Hedge, the objective function changes, thus the effective depth of the network changes

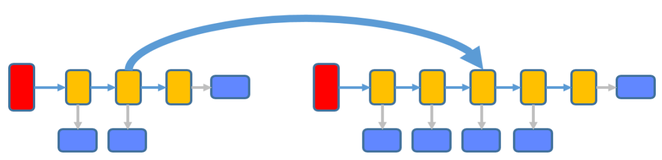
Smoothing Parameter is used to trade-off exploration and exploitation of model selection. It also facilitates continuous learning of all classifiers.

$$\alpha^{(l)} \leftarrow \max\left(\alpha^{(l)}, \frac{s}{L}\right)$$

Tackling Convergence Issues



Tackling Model Selection



By evaluating the online performance of the classifiers the weights α are adjusted automatically during the training procedure and the effective depth of the model is varied automatically in a data-drive manner

Parallel Interpretations

Some closely related ideas include:

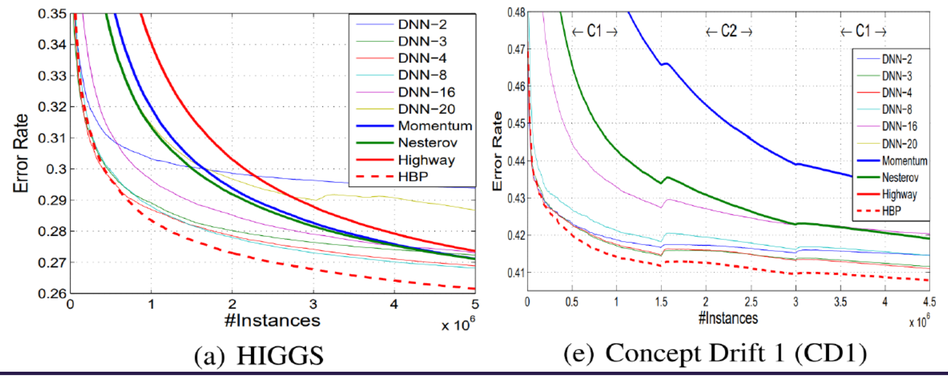
- Student-Teacher Learning (to get good initialization)
- Lifelong Learning (learn better as more data is available)
- Addressing Concept Drift using DNNs

Experiments and Insights

Convergence Behaviour on Stationary and Concept Drifting Datasets

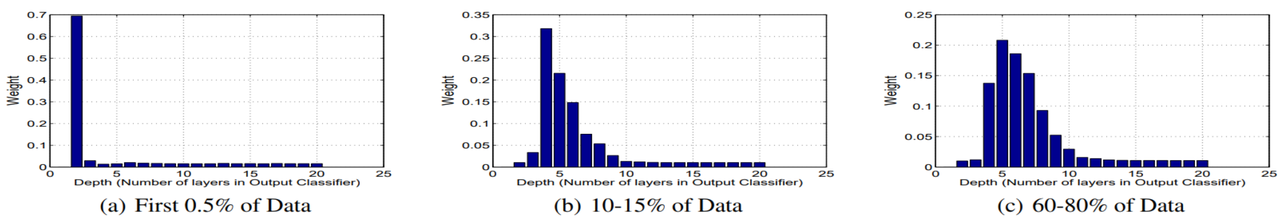
- HBP converges quickly, and matches performance of shallow networks
- It exploits depth better at a later stage & gets best overall performance

Baselines: Linear and Kernel OL | DNNs with varying depth | DNN-20 – with momentum, Highway
Proposed: Online Deep Learning by Hedge Backpropagation (DNN-20)

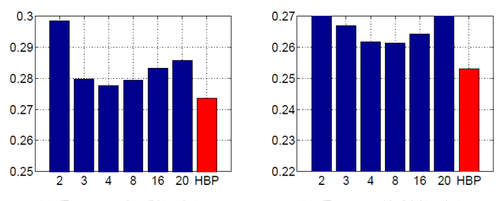


(a) HIGGS (e) Concept Drift 1 (CD1)

Varying the effective depth of the network during the course of online learning



(a) First 0.5% of Data (b) 10-15% of Data (c) 60-80% of Data



(a) Error in 10-15% of data (b) Error in 60-80% of data

HBP is able to perform better than all networks of different depth during different stages of learning

Error rate for varying depth of the networks: HBP's performance is relatively very stable regardless of depth of base network

Depth	12	16	20	30
Online BP	0.2692	0.2731	0.2868	0.4770
HBP	0.2609	0.2613	0.2615	0.2620

References and Code

1. Hoi, Steven CH, Jialei Wang, and Peilin Zhao. "Libol: A library for online learning algorithms." *JMLR 2014*
2. Hoi, Steven CH, Doyen Sahoo, Jing Lu, and Peilin Zhao. "Online Learning: A Comprehensive Survey." *arXiv preprint arXiv:1802.02871* (2018).
3. Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences* 55.1 (1997): 119-139.

LIBOL
Library for Online Learning Algorithms
Singapore http://libol.stevenhoi.org/
github.com/LIBOL/ODL

NATIONAL RESEARCH FOUNDATION
PRIME MINISTER'S OFFICE
SINGAPORE

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.