Robust deep learning in real world

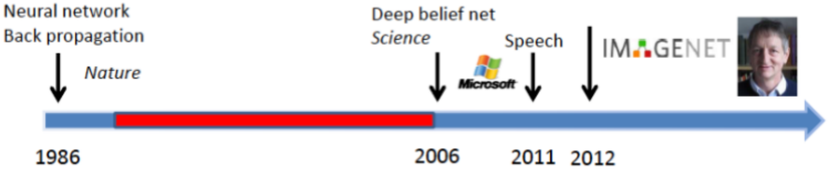
Nicolas Thome - Prof. at Cnam Paris
CEDRIC Lab, MSDMA Team

Artificial Intelligence Seminar
Mathematics and Computer Science (MICS) Lab
Centrale Supélec Paris Saclay

January 27, 2020

Deep Learning Success since 2010

- 90's / 2000's: difficult to train large deep models on existing databases

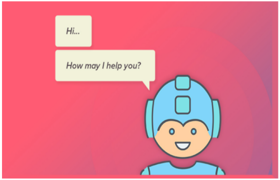


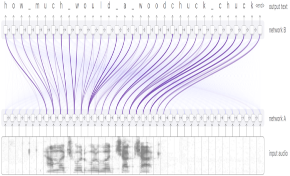
- **ILSVRC'12: the deep revolution**
  ⇒ outstanding success of ConvNets [Krizhevsky et al., 2012]

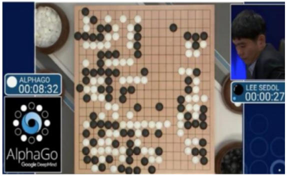
Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

Deep Learning everywhere since 2012

- Image classification, speech recognition
- chatbots, translation,
- Games, robotics







- **The formal Neuron**

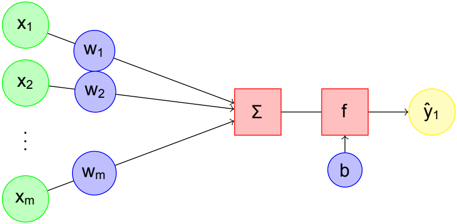
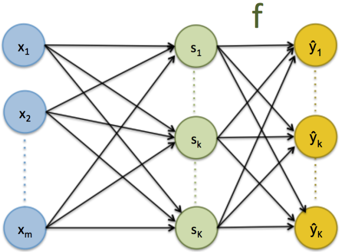


Figure: The formal neuron – **Credits: R. Herault**

x_i: inputs
w_i, b: weights
f: activation function
y: output of the neuron

$$y = f\left(w^\top x + b\right)$$

- <u>**Neural Networks:**</u> Stacking several formal neurons ⇒ **Perceptron**



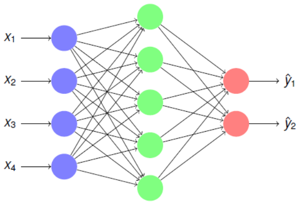
- **Soft-max Activation**:

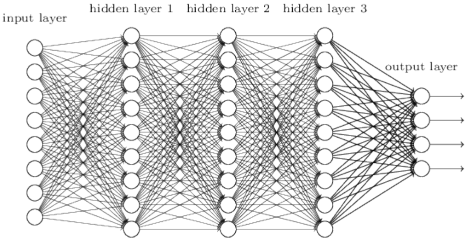
$$\hat{y_k} = f\left(s_k\right) = \frac{e^{s_k}}{\sum\limits_{k'=1}^{K} e^{s_{k'}}}$$

⇒ **Logistic Regression (LR) Model !**

Deep Neural Networks (DNN)

- **Multi-Layer Perceptron (MLP):** Stacking layers of neural networks
  - More complex and rich functions / Logistic Regression (LR)
  - **Neural network with one single hidden layer ⇒ universal approximator** [Cybenko, 1989]

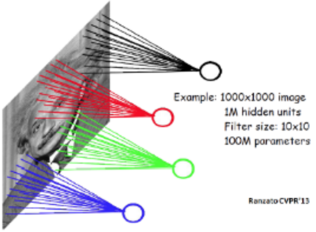


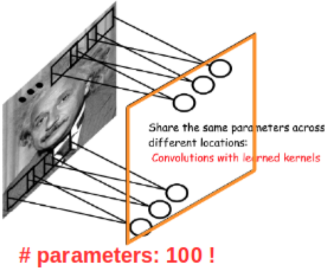


- **Basis of the "deep learning" field**
  - **Hidden layers: intermediate representations from data**
  - **Can be learned with Backpropagation algorithm [Lecun, 1985, Rumelhart et al., 1986]** (chain rule)

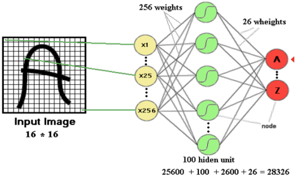
Convolutional Neural Networks (ConvNets)

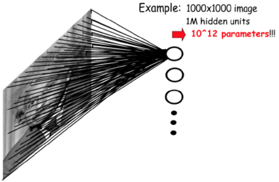
- **ConvNets:** sparse connectivity + shared weights





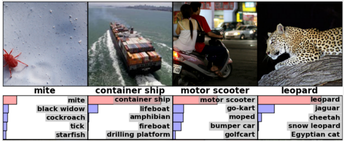
- **Local feature extraction ($\neq$ FCN)**
- Overcome parameter explosion for FCN on images



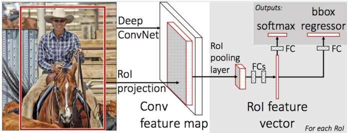


Deep Learning in Computer Vision

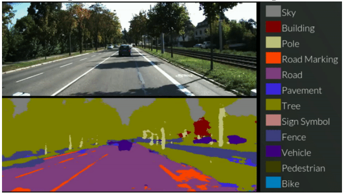
[*Krizhevsky*, 2012]



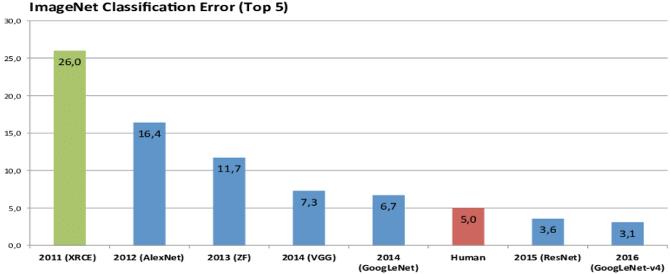
[*Girshick et al*. Fast R-CNN, 2015]



[*Kendall et al*. SegNet, 2015]

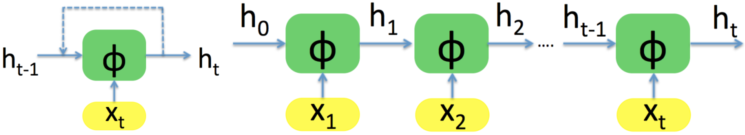


Brought significant improvements in multiple vision tasks



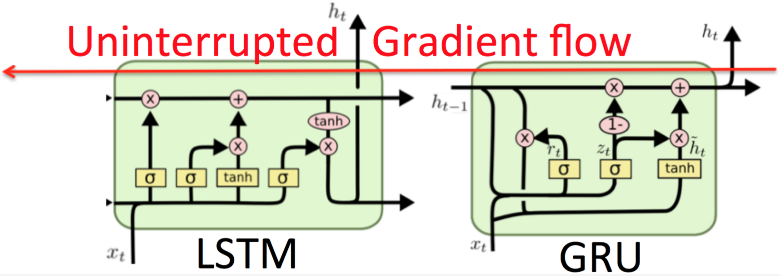
Recurrent Neural Networks (RNNs)

- **RNN Cell:** $\mathbf{h}_t = \phi(\mathbf{x}_t, \mathbf{h}_{t-1}) = f(\mathbf{U}\mathbf{x}_\mathbf{t} + \mathbf{W}\mathbf{h}_{\mathbf{t-1}} + \mathbf{b}_\mathbf{h})$ [Elman, 1990]
  - **$\mathbf{h}_t$: network memory up to time t $\Rightarrow$ Sequence processing**



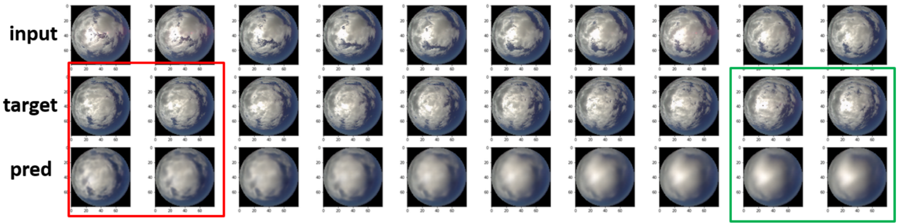
Folded RNN Unfolded RNN

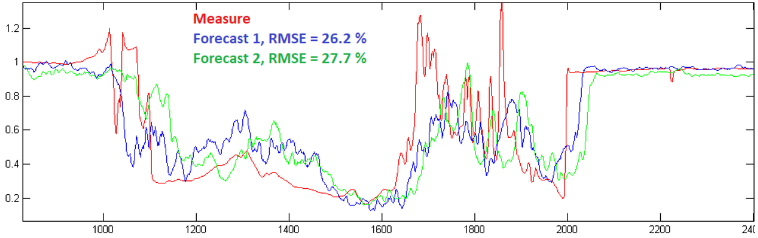
- **Specific architectures for vanishing gradients:**
  LSTM [Hochreiter and Schmidhuber, 1997], GRU [Cho et al., 2014]



Deep Learning for Sequence Processing

- RNNs SOTA for many sequential decision making tasks: speech, translation, text/music generation, times series, etc
- Ex: forecasting future frames for energy regulation (EDF)





Deep Learning Robustness

Deep Learning: huge gain in average performance, *e.g.* precision for classification, ℓ_2 loss for regression

- In several contexts, need to **optimize domain-specific metrics**
  $\Rightarrow$ **new DILATE loss for deep time series forecasting**
- Need for **performance certification in safety-critical applications: robustness**
  $\Rightarrow$ **new confidence / uncertainty measure for deep models**

[Evtimov et al., 2017]

Outline

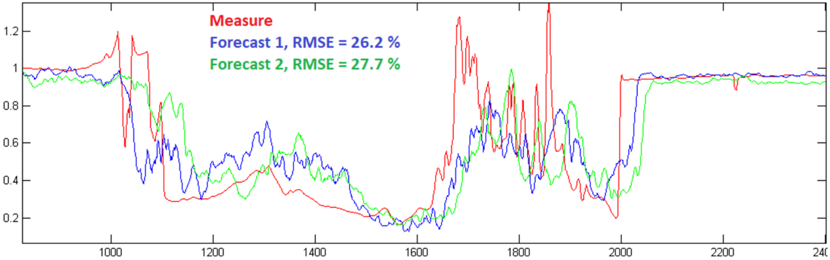
1. DILATE loss for training deep forecasting models
2. ConfidNet for confidence estimation

Context

Goal: Time series forecasting

- **multi-step** setting
- **non stationary** time series, that can present abrupt changes

Why ?: Important in many contexts, e.g. electricity (anticipate future drops of production), etc...



Traditional methods:

- Auto-Regressive models (ARMA, ARIMA,...) [Box et al., 2015]
- State Space Models (Exponential smoothing, ...) [Hyndman et al., 2008]

– Assumption: stationary time series

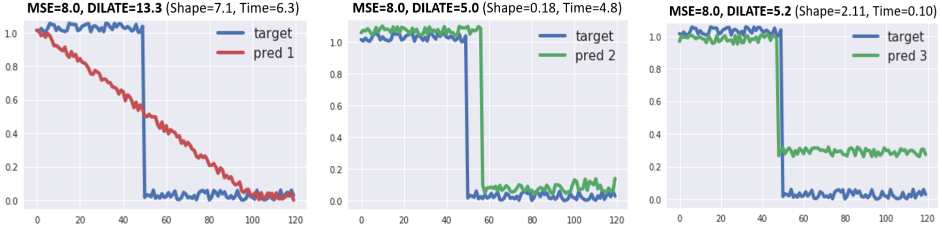
Deep learning models:

- Seq2Seq Recurrent Neural Networks [Yu et al., 2017b]
- Complex architectures for multivariate forecasting: attention mechanisms, tensor factorizations [Yu et al., 2016]
- Deep State Space Models for modeling uncertainty [Rangapuram et al., 2018]

... but all models are trained with the Mean Squared Error (MSE) !

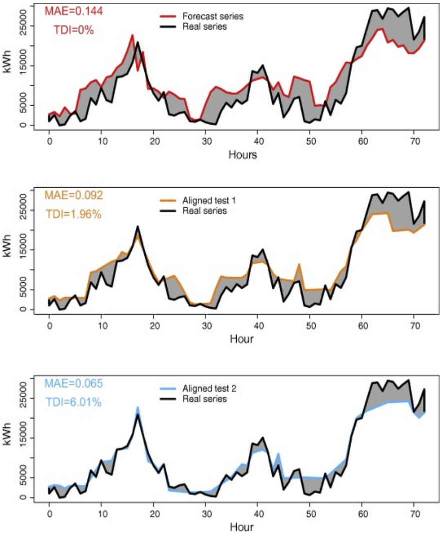
Motivation: MSE Loss Limitation

- MSE loss typically used for training forecasting problems not adapted to judge the quality of a forecast.



Non informative prediction | Correct shape, time delay | Correct time, inaccurate shape

Specific Metric for time series forecasting



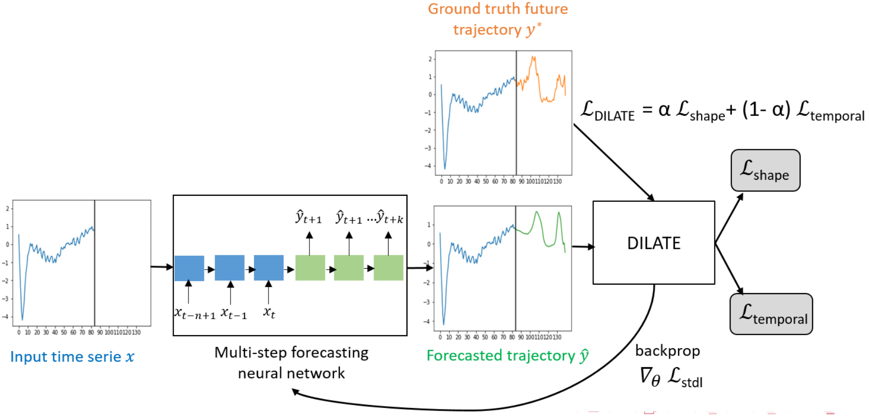
- Change Point Detection
  [Chang et al., 2019, Li et al., 2015]
- Hausdorff distance
  [Garreau et al., 2018, Truong et al., 2019]
- Ramp score
  [Florita et al., 2013, Vallance et al., 2017]
- Time Distrosion Index (TDI)
  [Frías-Paredes et al., 2017]

... but not differentiable! How to train deep models?

Proposal: DIstortion Loss with shApe and TimE (DILATE)

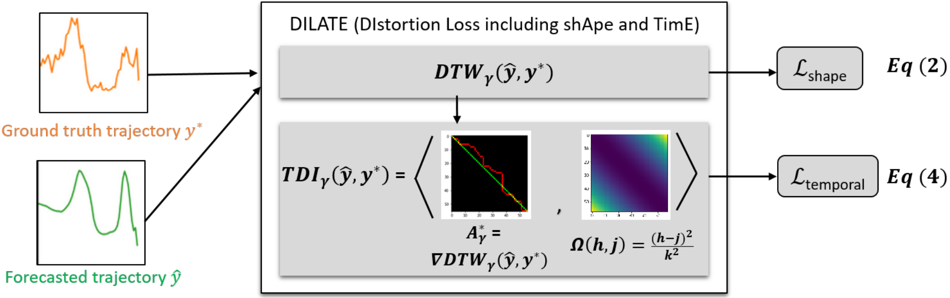
- Training dataset: N input time series $\mathcal{A} = \{\mathbf{x}_i\}_{i \in \{1:N\}}$
  - $\mathbf{x}_i = (\mathbf{x}_i^1, \ldots, \mathbf{x}_i^n) \in \mathbb{R}^{p \times n}$ input of length n
  - $\overset{*}{\mathbf{y}}_i = (\overset{*}{\mathbf{y}}_i^{1}, \ldots, \overset{*}{\mathbf{y}}_i^{k})$ GT output of length k
  - $\hat{\mathbf{y}}_i = (\hat{\mathbf{y}}_i^1, \ldots, \hat{\mathbf{y}}_i^k) \in \mathbb{R}^{d \times k}$ predicted output of length k (deep forecasting model)

$$\mathcal{L}_{DILATE}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \alpha \ \mathcal{L}_{shape}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) + (1 - \alpha) \ \mathcal{L}_{temporal}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \quad (1)$$



Training deep forecasting models with DILATE

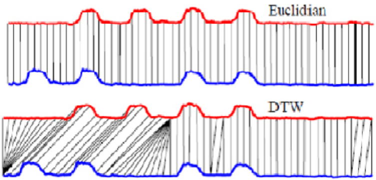
- $\mathcal{L}_{shape}$ and $\mathcal{L}_{temporal}$ based on Dynamic Time Warping [Sakoe and Chiba, 1990]



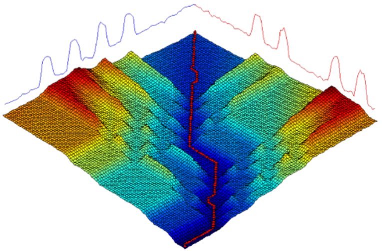
- $\mathcal{L}_{shape}$ and $\mathcal{L}_{temporal}$ differentiable wrt network parameters

Dynamic Time Warping (DTW) [Sakoe and Chiba, 1990]

- DTW: alignment between 2 time series: $DTW(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \min_{\mathbf{A} \in \mathcal{A}_{k,k}} \left\langle \mathbf{A}, \boldsymbol{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle$
- $\mathcal{A}_{k,k} \subset \{0,1\}^{k \times k}$: alignment paths (binary matrices), authorized moves $\rightarrow, \downarrow, \searrow$
- $\boldsymbol{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) := [\delta(\hat{\mathbf{y}}_i^h, \overset{*}{\mathbf{y}}_i^j)]_{h,j}$ pairwise cost matrix, *e.g.* $\delta(\hat{\mathbf{y}}_i^h, \overset{*}{\mathbf{y}}_i^j) = (\hat{\mathbf{y}}_i^h - \overset{*}{\mathbf{y}}_i^j)^2$



MSE vs DTW loss



Pairwise cost matrix and optimal alignment

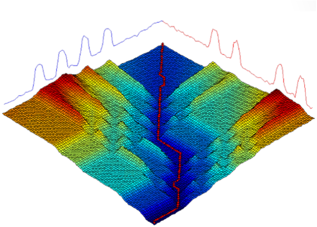
- ⊕ DTW good candidate for a shape loss
- ⊖ Not differentiable wrt $\boldsymbol{\Delta}$ ...

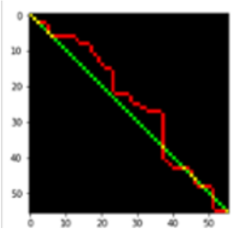
Shape term $\mathcal{L}_{shape}$ and Temporal term $\mathcal{L}_{temporal}$

- Soft min operator: $\min_\gamma(a_1, ..., a_n) = -\gamma \log(\sum_{i=1}^{n} \exp(-\frac{a_i}{\gamma}))$, $\gamma > 0$
- Soft-DTW [Cuturi and Blondel, 2017] for shape term:

$$\mathcal{L}_{shape}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = DTW_\gamma(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) := -\gamma \log\left(\sum_{\mathbf{A} \in \mathcal{A}_{k,k}} \exp\left(-\frac{\left\langle \mathbf{A}, \boldsymbol{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i)\right\rangle}{\gamma}\right)\right) \quad (2)$$

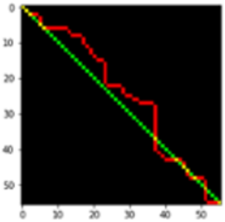
- Temporal term: based on DTW optimal path $\mathbf{A}^* = \underset{A \in \mathcal{A}_{k,k}}{\text{argmin}} \left\langle \mathbf{A}, \boldsymbol{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i)\right\rangle$:
  - A^* along the main diagonal $\Rightarrow$ no temporal distortion
  - A^* departs from the diagonal $\Rightarrow$ presence of temporal distortion





Temporal term $\mathcal{L}_{temporal}$

- Generalized Time Distortion Index (TDI) [Frías-Paredes et al., 2017]



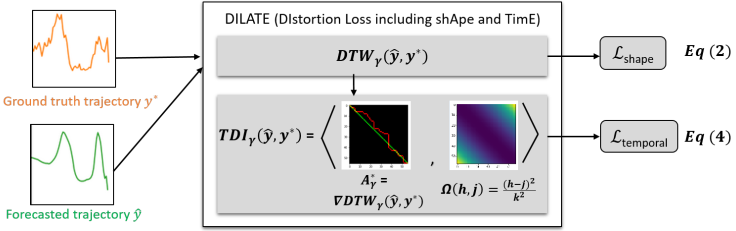
$$TDI(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = \langle \mathbf{A}^*, \mathbf{\Omega} \rangle = \left\langle \underset{\mathbf{A} \in \mathcal{A}_{k,k}}{\arg\min} \left\langle \mathbf{A}, \mathbf{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle, \mathbf{\Omega} \right\rangle \quad (3)$$

- $\mathbf{\Omega}$: penalizing matrix of size $k \times k$, *e.g.* $\mathbf{\Omega}(h,j) = \frac{1}{k^2}(h-j)^2$

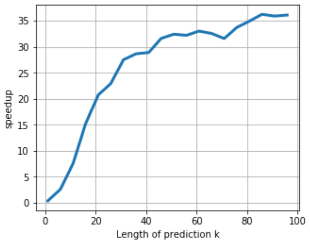
- $\mathbf{A}^* = \nabla_\Delta \text{DTW}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i)$ not differentiable
- $\mathbf{A}^* \approx \mathbf{A}^*_\gamma = \nabla_\Delta DTW_\gamma(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) = 1/Z \ \sum_{\mathbf{A} \in \mathcal{A}_{k,k}} \mathbf{A} \exp^{-\frac{\left\langle \mathbf{A}, \mathbf{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle}{\gamma}}$
- **Smooth temporal loss:** $\mathcal{L}_{temporal}$

$$\mathcal{L}_{temporal}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) := \left\langle \mathbf{A}^*_\gamma, \mathbf{\Omega} \right\rangle = \frac{1}{Z} \sum_{\mathbf{A} \in \mathcal{A}_{k,k}} \langle \mathbf{A}, \mathbf{\Omega} \rangle \exp^{-\frac{\left\langle \mathbf{A}, \mathbf{\Delta}(\hat{\mathbf{y}}_i, \overset{*}{\mathbf{y}}_i) \right\rangle}{\gamma}} \quad (4)$$

Training deep forecasting models with DILATE



- Direct computation of $\mathcal{L}_{shape}$ and $\mathcal{L}_{temporal}$ intractable ($|\mathcal{A}_{k,k}| = O(exp(k^2))$)
- Solution: dynamic programming $\Rightarrow$ custom forward/backward implementation



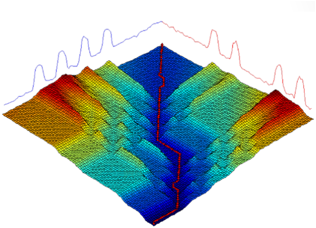
Variants of DILATE

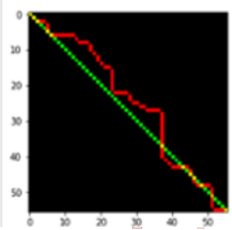
- DILATE-t: "tangled" variant of DILATE

DILATE	$\min_{A}{}_{\gamma} \langle \mathbf{A}, \mathbf{\Delta} \rangle + \langle A^*, \mathbf{\Omega} \rangle$
DILATE-t	$\min_{A}{}_{\gamma} \langle \mathbf{A}, \mathbf{\Delta} + \mathbf{\Omega} \rangle$

- DILATE-t: penalization matrix $\mathbf{\Omega}$ inside the minimization of DTW
  - Shape and temporal term mixed during minimization
- DILATE-t subsumes well-known temporally-constrained DTW methods:

Sakoe-Chiba hard band constraint	$\Omega(h,j) = +\infty$ if $\|h-j\| > T$, 0 otherwise
Weighted DTW	$\Omega(h,j) = f(\|i-j\|)$, f increasing function





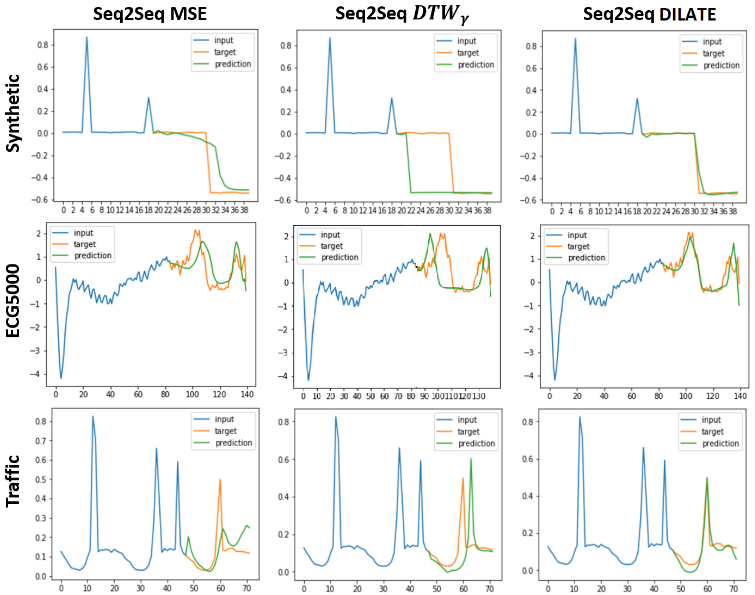
Experiments

Experimental setup: evaluate the k-step future trajectories

3 non stationary datasets from various domains:

- Synthetic
- ECG5000
- Traffic

Qualitative forecasting results



Quantitative results

Training with DILATE vs MSE leads to:

- Equivalent results evaluated on MSE
- Better results evaluated on shape (DTW)
- Better results evaluated on timing (TDI)

		Fully connected network (MLP)			Recurrent neural network (Seq2Seq)		
Dataset	Eval	MSE	DTW_γ	DILATE (ours)	MSE	DTW_γ	DILATE (ours)
Synth	MSE	**1.65** ± **0.14**	4.82 ± 0.40	**1.67**± **0.184**	**1.10** ± **0.17**	2.31 ± 0.45	**1.21** ± **0.13**
	DTW	38.6 ± 1.28	**27.3** ± **1.37**	32.1 ± 5.33	**24.6** ± **1.20**	**22.7** ± **3.55**	**23.1** ± **2.44**
	TDI	15.3 ± 1.39	26.9 ± 4.16	**13.8** ± **0.712**	17.2 ± 1.22	20.0 ± 3.72	**14.8** ± **1.29**
ECG	MSE	**31.5** ± **1.39**	70.9 ± 37.2	37.2 ± 3.59	**21.2** ± **2.24**	75.1 ± 6.30	30.3 ± 4.10
	DTW	19.5 ± 0.159	18.4 ± 0.749	**17.7** ± **0.427**	17.8 ± 1.62	17.1 ± 0.650	**16.1** ± **0.156**
	TDI	**7.58** ± **0.192**	38.9 ± 8.76	**7.21** ± **0.886**	8.27 ± 1.03)	27.2 ± 11.1	**6.59** ± **0.786**
Traffic	MSE	**0.620** ± **0.010**	2.52 ± 0.230	1.93 ± 0.080	**0.890** ± **0.11**	2.22 ± 0.26	**1.00** ± **0.260**
	DTW	24.6 ± 0.180	**23.4** ± **5.40**	**23.1** ± **0.41**	24.6 ± 1.85	**22.6** ± **1.34**	**23.0** ± **1.62**
	TDI	**16.8** ± **0.799**	27.4 ± 5.01	**16.7** ± **0.508**	**15.4** ± **2.25**	22.3 ± 3.66	**14.4**± **1.58**

Table: Forecasting results evaluated with MSE, Shape and Time metrics, averaged over 10 runs (mean ± standard deviation). For each experiment, best method(s) (Student t-test) in bold.

Evaluation with external metrics

- Shape: **ramp score** [Vallance et al., 2017]
- Time: **Hausdorff distance** between 2 sets of change points

		MSE	DTW_γ	DILATE (ours)
Synthetic	Hausdorff	2.87 ± 0.127	3.45 ± 0.318	**2.70 ± 0.166**
	Ramp score (x10)	5.80 ± 0.104	**4.27 ± 0.800**	4.99 ± 0.460
ECG5000	Hausdorff	**4.32 ± 0.505**	6.16 ± 0.854	**4.23 ± 0.414**
	Ramp score	**4.84 ± 0.240**	**4.79 ± 0.365**	**4.80 ± 0.249**
Traffic	Hausdorff	**2.16 ± 0.378**	**2.29 ± 0.329**	**2.13 ± 0.514**
	Ramp score (x10)	6.29 ± 0.319	**5.78 ± 0.404**	**5.93 ± 0.235**

Table: Forecasting results of Seq2Seq evaluated with Hausdorff and Ramp Score, averaged over 10 runs (mean ± standard deviation). For each experiment, best method(s) (Student t-test) in bold.

Comparison to tangled variants of DILATE

Eval loss		DILATE (ours)	DILATE^t-Weighted	DILATE^t-Band Constraint
Euclidian	MSE (x100)	**1.21** ± **0.130**	1.36 ± 0.107	1.83 ± 0.163
Shape	DTW (x100)	**23.1** ± **2.44**	**20.5** ± **2.49**	**21.6** ± **1.74**
	Ramp	**4.99** ± **0.460**	**5.56** ± **0.87**	**5.23** ±**0.439**
Time	TDI (x10)	**14.8** ± **1.29**	17.8 ± 1.72	19.6 ± 1.72
	Hausdorff	**2.70** ± **0.166**	**2.85** ± **0.210**	3.30 ± 0.273

Table: Comparison to the tangled variants of DILATE for the Seq2Seq model on the Synthetic dataset, averaged over 10 runs (mean ± standard deviation).

State of the art comparison

Baselines:

- LSTNet [Lai et al., 2018]: mono-step model, applied recursively for multi-step
- Deep AR [Laptev et al., 2017]: trained with MSE
- TT-RNN [Yu et al., 2017a]: SOTA Seq2Seq model

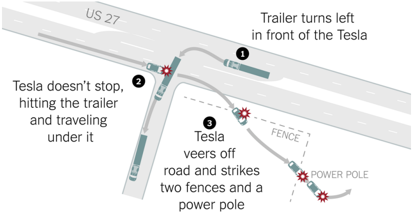
Eval loss		LSTNet-rec (MSE)	TT-RNN (MSE)	Deep AR (MSE)	Seq2Seq (DILATE)	TT-RNN (DILATE)
Euclidian	MSE	1.74 ± 0.11	**0.840 ± 0.106**	0.966 ± 0.068	1.00 ± 0.260	**0.930 ± 0.09**
Shape	DTW	42.0 ± 2.2	25.9 ± 1.99	27.8 ± 1.55	23.0 ± 1.62	**21.4 ± 0.79**
	Ramp	9.00 ± 0.577	6.71 ± 0.546	7.56 ± 0.42	5.93 ± 0.235	**5.27 ± 0.27**
Time	TDI	25.7 ± 4.75	17.8 ± 1.73	**14.6 ± 0.94**	**14.4 ± 1.58**	**15.7 ± 1.02**
	Hausdorff	2.34 ± 1.41	2.19 ± 0.12	2.04 ± 0.11	2.13 ± 0.514	**1.88 ± 0.153**

$\Rightarrow$ DILATE can improve the performance of SOTA multi-step architecture on shape and time metrics, and equivalent on MSE

Outline

Robustness issues

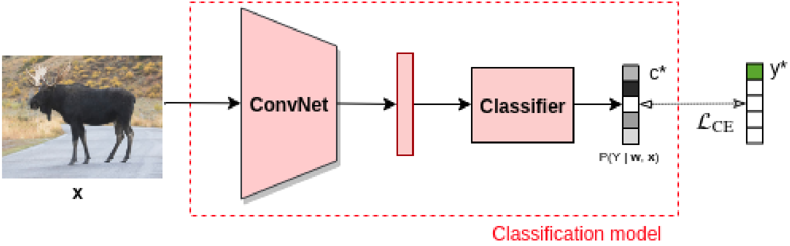
Tesla's car crash back in 2016, due to a confusion between white side of trailer and brightly lit sky



⇒ **Are neural network's predictions reliable? How much is the model certain about our output? How do we account for uncertainty?**

Confidence Estimation in Deep Learning

Classification framework
$\mathcal{D} = \{(\mathbf{x}_i, y_i^*)\}_{i=1}^N$ with $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i^* \in \mathcal{Y} = \{1, ..., K\}$.
One can infer predicted class $\hat{y} = \text{argmax}_{k\in\mathcal{Y}}\, p(Y = k|\mathbf{w}, \mathbf{x})$.

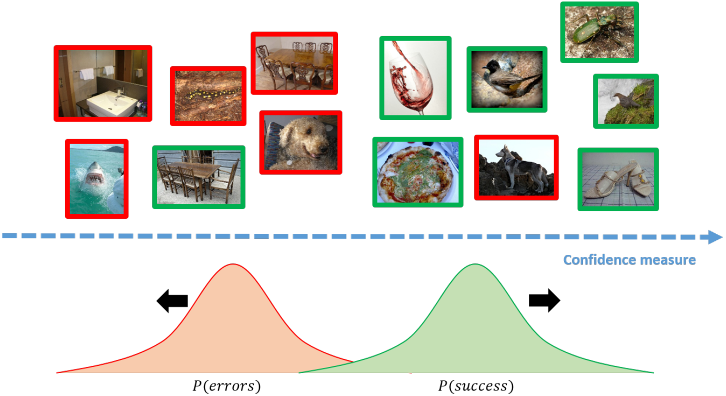


- **Maximum Class Probability** [Hendrycks and Gimpel, 2017]
  A confidence measure baseline for deep neural networks:

$$\mathrm{MCP}(\mathbf{x}) = \max_{k\in\mathcal{Y}} p(Y = k|\mathbf{w}, \mathbf{x})$$

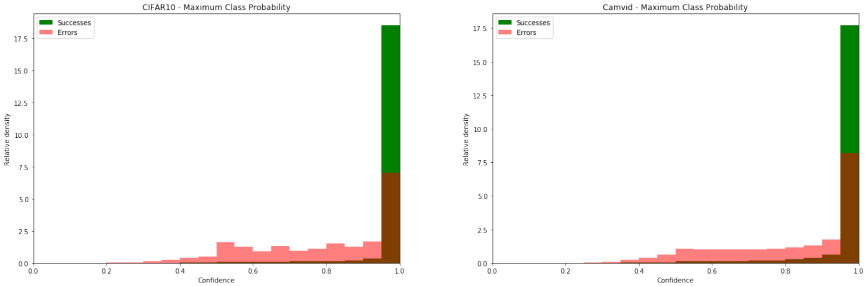
Goal

Provide **reliable confidence measures** over model's predictions whose ranking among samples enables to **distinguish correct from erroneous predictions**.



MCP, a sub-optimal ranking confidence measure

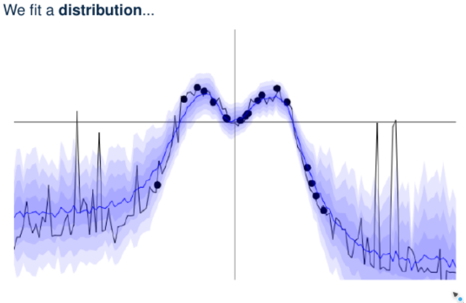
$$\mathrm{MCP}(\mathbf{x}) = \max_{k \in \mathcal{Y}} p(Y = k | \mathbf{w}, \mathbf{x})$$

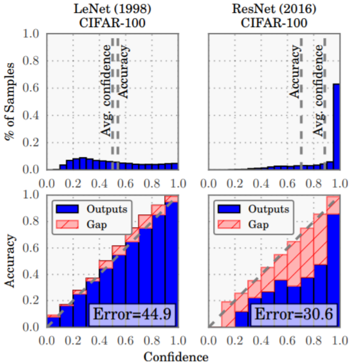


- **overlapping distributions** between successes vs. errors
  $\Rightarrow$ hard to distinguish

Beyond MCP: Related Works

- Bayesian deep learning, *e.g.* MC-Dropout [Gal and Ghahramani, 2016]
- Specific confidence criterion for failure prediction, *e.g.* Trust Score [Jiang et al., 2018]
- Calibration related to overconfident prediction [Guo et al., 2017, Neumann et al., 2018]

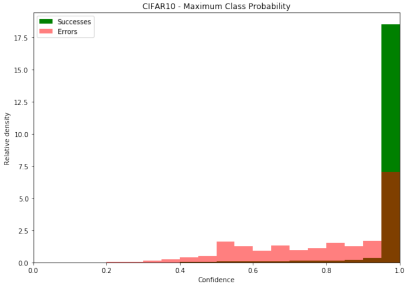


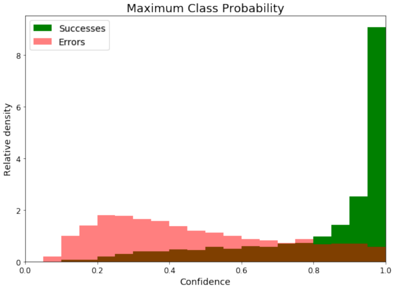


MCP, a sub-optimal ranking confidence measure

$$\mathrm{MCP}(\mathbf{x}) = \max_{k \in \mathcal{Y}} p(Y = k | \mathbf{w}, \mathbf{x})$$

- Overconfident prediction values
  ⇒ calibration [Guo et al., 2017, Neumann et al., 2018]
- BUT: calibration does not change error/correct prediction rankings





Our Approach: True Class Probability

When missclassifying, MCP $\Leftrightarrow$ probability of the wrong class.
$\Rightarrow$ **what if we had taken the probability of the true class?**

True Class Probability

Given a sample $(\mathbf{x}, y^*)$ and a model parametrized by $\mathbf{w}$, *True Class Probability* writes as:

$$\mathrm{TCP}(\mathbf{x}, y^*) = p(Y = y^* | \mathbf{w}, \mathbf{x})$$

Theoretical guarantees:

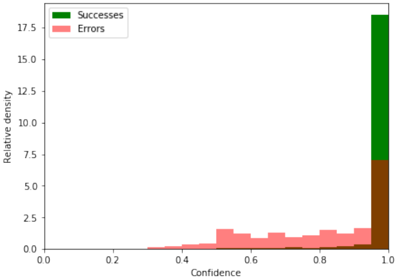
- $\mathrm{TCP}(\mathbf{x}, y^*) > 1/2 \Rightarrow \hat{y} = y^*$
- $\mathrm{TCP}(\mathbf{x}, y^*) < 1/K \Rightarrow \hat{y} \neq y^*$

N.B: a normalized variant present stronger guarantees:

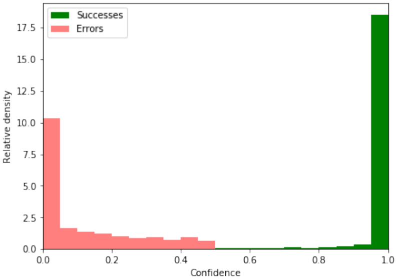
$$TCP^r(\mathbf{x}, y^*) = \frac{p(Y = y^* | \mathbf{w}, \mathbf{x})}{p(Y = \hat{y} | \mathbf{w}, \mathbf{x})}$$

TCP, a reliable confidence criterion

VGG16 on CIFAR-10



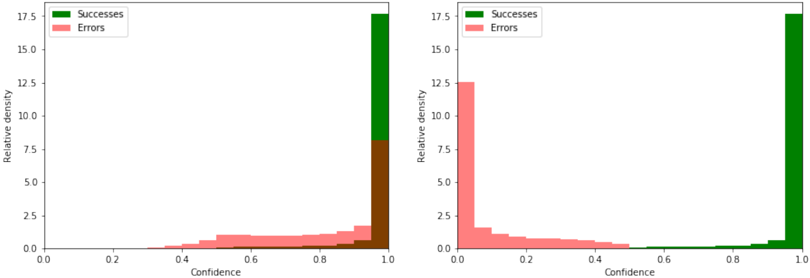
(a) Maximum Class Probability



(b) Our Proposal (True Class Probability)

TCP, a reliable confidence criterion

SegNet on CamVid



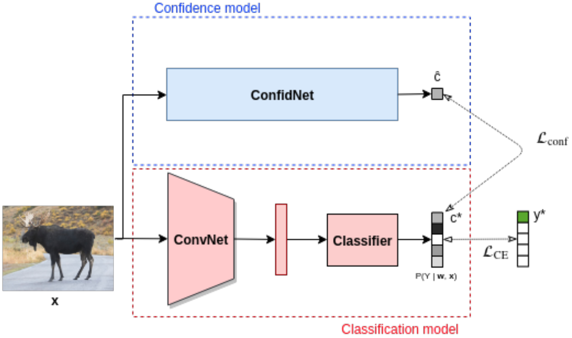
(a) Maximum Class Probability

(b) Our Proposal (True Class Probability)

ConfidNet: Learning TCP Model Confidence

However, $TCP(\mathbf{x}, y^*)$ is **unknown** at test time.

Given $\mathcal{D}_{train}$, **learn a confidence model** with parameters θ such that $\forall \mathbf{x} \in \mathcal{D}_{train}$, its scalar output $\hat{c}(\mathbf{x}, \theta)$ close to $TCP(\mathbf{x}, y^*)$

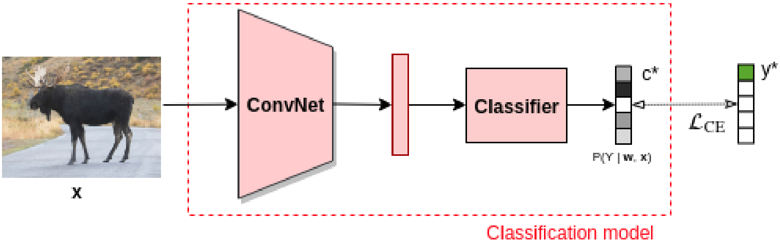


As $TCP(\mathbf{x}, y^*) \in [0, 1]$, we propose ℓ_2 loss to train ConfidNet:

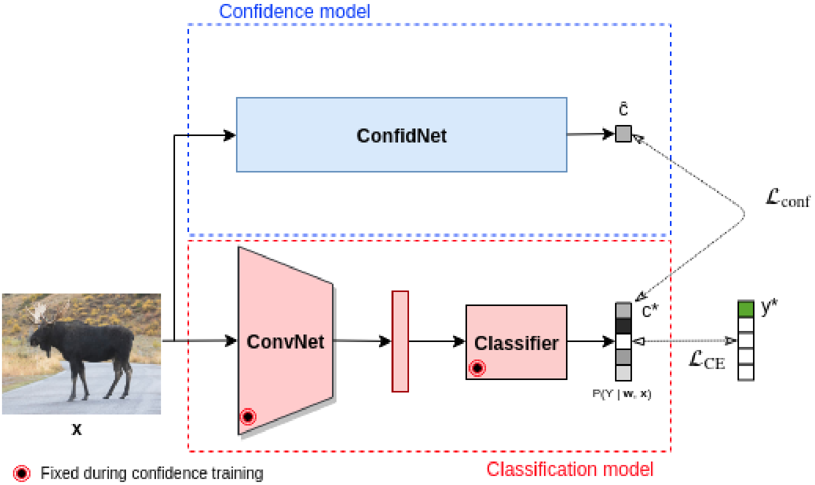
$$\mathcal{L}_{\text{conf}}(\theta; \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} (\hat{c}(\mathbf{x}_i, \theta) - c^*(\mathbf{x}_i, y_i^*))^2$$

N.B**: $c^*(\mathrm{x}, y^*) = TCP(\mathrm{x}, y^*)$ **(or** $TCP^r(\mathrm{x}, y^*)$**)

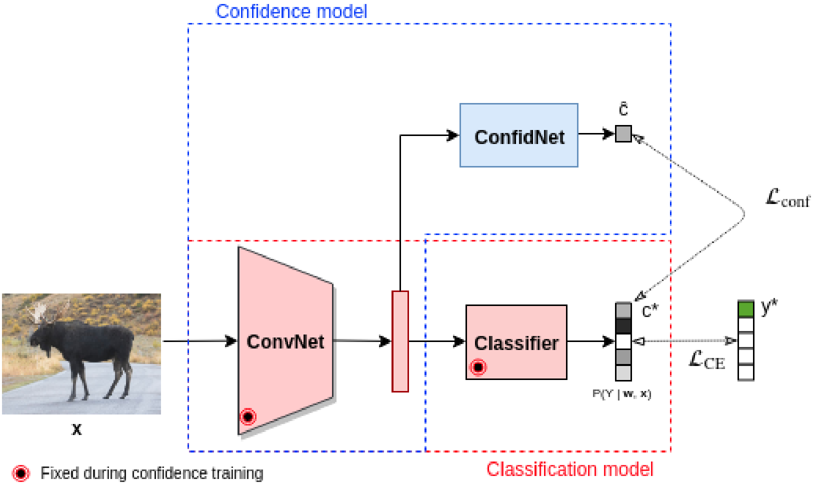
ConfidNet learning scheme



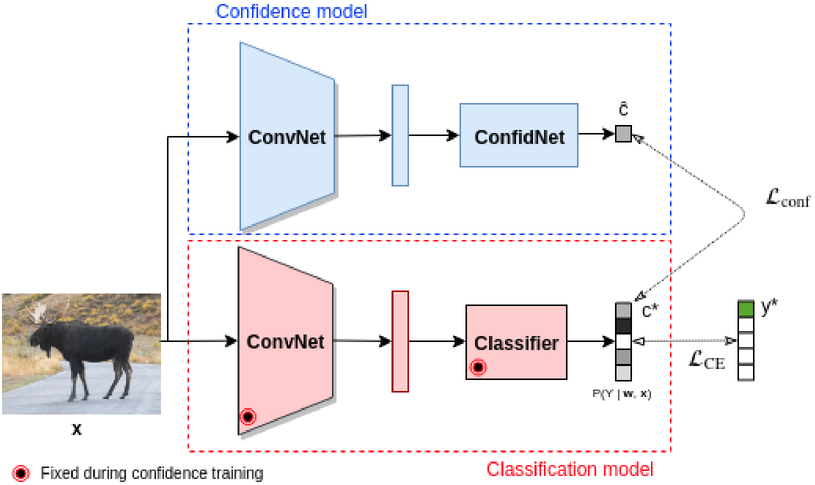
ConfidNet learning scheme



Efficient ConfidNet learning scheme (1/2)



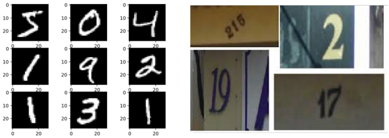
Efficient ConfidNet learning scheme (2/2)

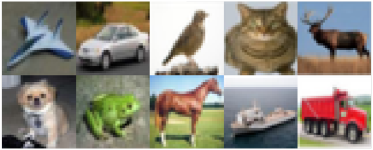


Experiments

Traditional public **image classification** and **semantic segmentation** datasets

- **MNIST**: 32x32 BW, 10 classes, 60K training + 10K test
- **SVHN**: 32x32 RGB , 10 classes, 73K training + 26K test
- **CIFAR-10 & CIFAR-100**: 32x32 RGB, *10 / 100 classes*, 50K training + 10K test
- **CamVid**: *semantic segmentation* , 360x480, 11 classes



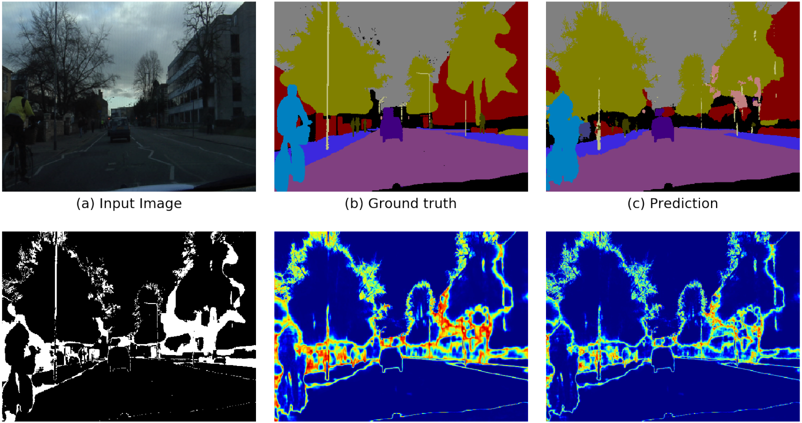


Quantitative results

Dataset	Model	FPR-95%-TPR	AUPR-Error	AUPR-Success	AUC
MNIST MLP	Baseline (MCP)	14.87	37.70	99.94	97.13
	MCDropout	15.15	38.22	99.94	97.15
	TrustScore	12.31	52.18	99.95	97.52
	ConfidNet (Ours)	**11.79**	**57.37**	**99.95**	**97.83**
MNIST Small ConvNet	Baseline (MCP)	5.56	35.05	99.99	98.63
	MCDropout	5.26	38.50	99.99	98.65
	TrustScore	10.00	35.88	99.98	98.20
	ConfidNet (Ours)	**3.33**	**45.89**	**99.99**	**98.82**
SVHN Small ConvNet	Baseline (MCP)	31.28	48.18	99.54	93.20
	MCDropout	36.60	43.87	99.52	92.85
	TrustScore	34.74	43.32	99.48	92.16
	ConfidNet (Ours)	**28.58**	**50.72**	**99.55**	**93.44**
CIFAR-10 VGG16	Baseline (MCP)	47.50	45.36	99.19	91.53
	MCDropout	49.02	46.40	**99.27**	92.08
	TrustScore	55.70	38.10	98.76	88.47
	ConfidNet (Ours)	**44.94**	**49.94**	99.24	**92.12**
CIFAR-100 VGG16	Baseline (MCP)	67.86	71.99	92.49	85.67
	MCDropout	64.68	72.59	**92.96**	86.09
	TrustScore	71.74	66.82	91.58	84.17
	ConfidNet (Ours)	**62.96**	**73.68**	92.68	**86.28**
CamVid SegNet	Baseline (MCP)	63.87	48.53	96.37	84.42
	MCDropout	62.95	49.35	96.40	84.58
	TrustScore		20.42	92.72	68.33
	ConfidNet (Ours)	**61.52**	**50.51**	**96.58**	**85.02**

Qualitative results

Failure detection for **semantic segmentation** on CamVid dataset

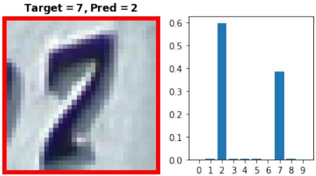


(a) Input Image (b) Ground truth (c) Prediction

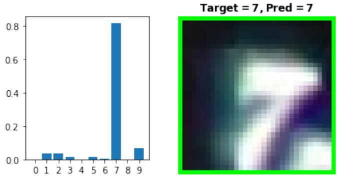
(d) Model Errors (e) ConfidNet (f) MCP

Qualitative results

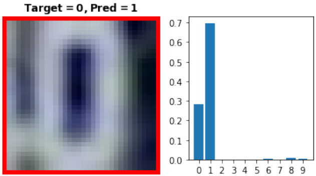
Entropy as a confident estimate, such as in
MC-Dropout [Gal and Ghahramani, 2016], may not always be adequate



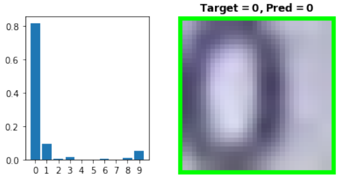
(a) MCP=0.596, MCDropout=-0.787, *ConfidNet*=0.449



(b) MCP=0.816, MCDropout=-0.786, *ConfidNet*=0.894



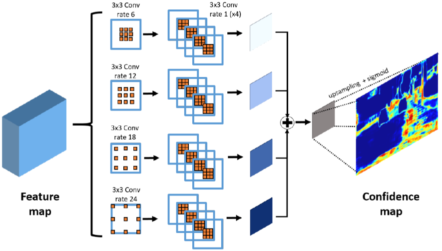
(c) MCP=0.696, MCDropout=-0.726, *ConfidNet*=0.436

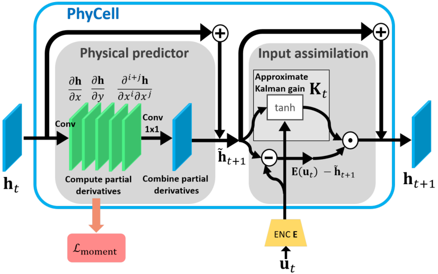


(d) MCP=0.814, MCDropout=-0.725, *ConfidNet*=0.886

Conclusion

- **DILATE & ConfidNet**: new loss & confidence for deep neural networks
  - Agnostic to model archi, data and tasks
- ConfidNet perspectives:
  - Application to Unsupervised Domain Adaptation (UDA)
  - Relative *vs* absolute confidence, out-of-distributions
- DILATE perspectives:
  - Deep archi with physical priors
  - Weakly-supervised predictions





Thank your for your attention!

- **DILATE: Vincent Le Guen, Nicolas Thome**
  - **NeurIPS'19 paper**: Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models
  - **GitHub code:** `https://github.com/vincent-leguen/DILATE`
- **ConfidNet: Charles Corbière, Nicolas Thome, Avner Bar-Hen, Matthieu Cord, Patrick Pérez**
  - **NeurIPS'19 paper**: Addressing Failure Prediction by Learning Model Confidence
  - **GitHub code:** `https://github.com/valeoai/ConfidNet`

References I

[Box et al., 2015] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015).
Time series analysis: forecasting and control.
John Wiley & Sons.

[Chang et al., 2019] Chang, W.-C., Li, C.-L., Yang, Y., and Póczos, B. (2019).
Kernel change-point detection with auxiliary deep generative models.
In *International Conference on Learning Representations (ICLR)*.

[Cho et al., 2014] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014).
Learning phrase representations using rnn encoder-decoder for statistical machine translation.
cite arxiv:1406.1078Comment: EMNLP 2014.

[Cuturi and Blondel, 2017] Cuturi, M. and Blondel, M. (2017).
Soft-dtw: a differentiable loss function for time-series.
In *International Conference on Machine Learning (ICML)*, pages 894–903.

[Cybenko, 1989] Cybenko, G. (1989).
Approximation by superpositions of a sigmoidal function.
Mathematics of control, signals and systems, 2(4):303–314.

[Elman, 1990] Elman, J. L. (1990).
Finding structure in time.
COGNITIVE SCIENCE, 14(2):179–211.

[Florita et al., 2013] Florita, A., Hodge, B.-M., and Orwig, K. (2013).
Identifying wind and solar ramping events.
In *2013 IEEE Green Technologies Conference (GreenTech)*, pages 147–152. IEEE.

[Frías-Paredes et al., 2017] Frías-Paredes, L., Mallor, F., Gastón-Romeo, M., and León, T. (2017).
Assessing energy forecasting inaccuracy by simultaneously considering temporal and absolute errors.
Energy Conversion and Management, 142:533–546.

References II

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016).
Dropout as a bayesian approximation: Representing model uncertainty in deep learning.
In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org.

[Garreau et al., 2018] Garreau, D., Arlot, S., et al. (2018).
Consistent change-point detection with kernels.
Electronic Journal of Statistics, 12(2):4440–4486.

[Guo et al., 2017] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017).
On calibration of modern neural networks.
In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 1321–1330.

[Hendrycks and Gimpel, 2017] Hendrycks, D. and Gimpel, K. (2017).
A baseline for detecting misclassified and out-of-distribution examples in neural networks.
Proceedings of International Conference on Learning Representations.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997).
Long short-term memory.
Neural Comput., 9(8):1735–1780.

[Hyndman et al., 2008] Hyndman, R., Koehler, A. B., Ord, J. K., and Snyder, R. D. (2008).
Forecasting with exponential smoothing: the state space approach.
Springer Science & Business Media.

[Jiang et al., 2018] Jiang, H., Kim, B., Guan, M., and Gupta, M. (2018).
To trust or not to trust a classifier.
In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 5541–5552. Curran Associates, Inc.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012).
Imagenet classification with deep convolutional neural networks.
In *Advances in neural information processing systems*, pages 1097–1105.

References III

[Lai et al., 2018] Lai, G., Chang, W.-C., Yang, Y., and Liu, H. (2018).
Modeling long-and short-term temporal patterns with deep neural networks.
In *ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM.

[Laptev et al., 2017] Laptev, N., Yosinski, J., Li, L. E., and Smyl, S. (2017).
Time-series extreme event forecasting with neural networks at Uber.
In *International Conference on Machine Learning (ICML)*, number 34, pages 1–5.

[Lecun, 1985] Lecun, Y. (1985).
Une procedure d'apprentissage pour reseau a seuil asymmetrique (A learning scheme for asymmetric threshold networks), pages 599–604.

[Li et al., 2015] Li, S., Xie, Y., Dai, H., and Song, L. (2015).
M-statistic for kernel change-point detection.
In *Advances in Neural Information Processing Systems (NIPS)*, pages 3366–3374.

[Neumann et al., 2018] Neumann, L., Zisserman, A., and Vedaldi, A. (2018).
Relaxed softmax: Efficient confidence auto-calibration for safe pedestrian detection.
In *Machine Learning for Intelligent Transportation Systems Workshop, NIPS*.

[Rangapuram et al., 2018] Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. (2018).
Deep state space models for time series forecasting.
In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7785–7794.

[Rumelhart et al., 1986] Rumelhart, D., Hinton, G., and Williams, R. (1986).
Learning representations by back-propagating errors.
Nature, 323:533–536.

[Sakoe and Chiba, 1990] Sakoe, H. and Chiba, S. (1990).
Dynamic programming algorithm optimization for spoken word recognition.
Readings in speech recognition, 159:224.

References IV

[Truong et al., 2019] Truong, C., Oudre, L., and Vayatis, N. (2019).
Supervised kernel change point detection with partial annotations.
In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3147–3151. IEEE.

[Vallance et al., 2017] Vallance, L., Charbonnier, B., Paul, N., Dubost, S., and Blanc, P. (2017).
Towards a standardized procedure to assess solar forecast accuracy: A new ramp and time alignment metric.
Solar Energy, 150:408–422.

[Yu et al., 2016] Yu, H.-F., Rao, N., and Dhillon, I. S. (2016).
Temporal regularized matrix factorization for high-dimensional time series prediction.
In *Advances in neural information processing systems (NIPS)*, pages 847–855.

[Yu et al., 2017a] Yu, R., Zheng, S., Anandkumar, A., and Yue, Y. (2017a).
Long-term forecasting using tensor-train RNNs.
arXiv preprint arXiv:1711.00073.

[Yu et al., 2017b] Yu, R., Zheng, S., and Liu, Y. (2017b).
Learning chaotic dynamics using tensor recurrent neural networks.
In *ICML Workshop on Deep Structured Prediction*, volume 17.