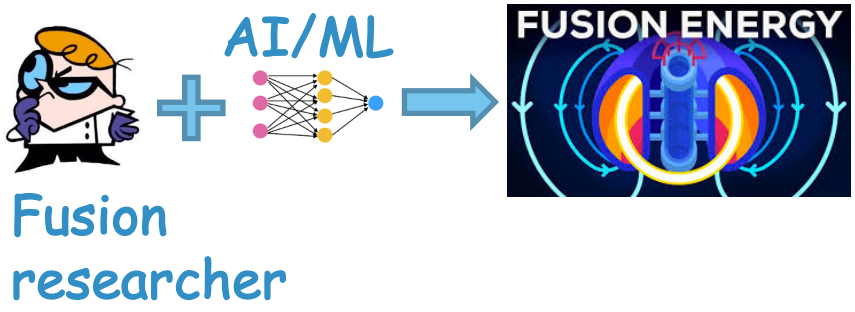
Accurate disruption prediction on the DIII-D tokamak using deep learning with raw, multi-scale diagnostic data

R. Michael Churchill (*PPPL*)

B. Tobias (*LANL*), Y. Zhu (*U.C. Davis*), J. Choi (*ORNL*), R. Kube (*PPPL*)

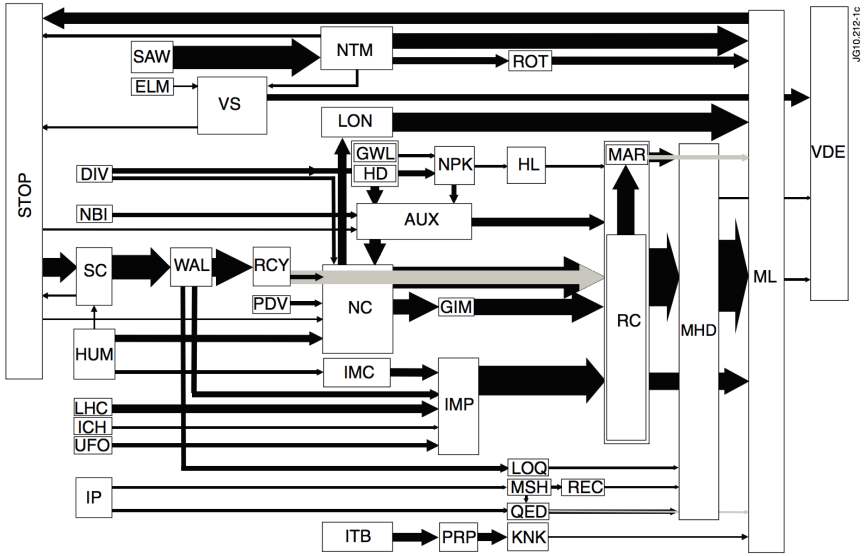


PPPL PRINCETON PLASMA PHYSICS LABORATORY

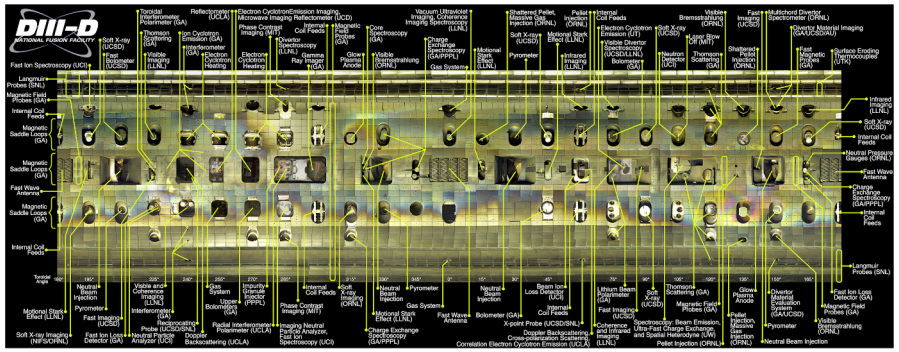
PRINCETON UNIVERSITY

Disruption prediction is made difficult due to multi-physics, multi-scale nature

- Many pathways can lead to a disruption occurring
- Data-driven prediction has focused on ~10 physics quantities
- Plethora of diagnostics on tokamak devices can be incorporated using deep learning to enhance disruption prediction capabilities



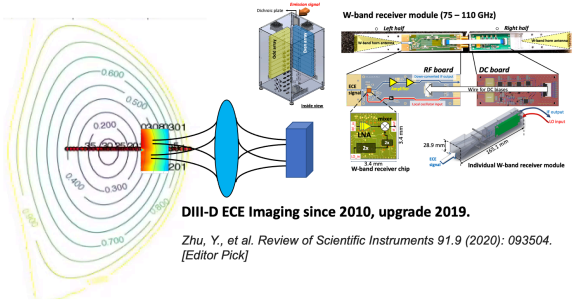
De Vries, P. C., et. al. (2011). Survey of disruption causes at JET. *Nuclear Fusion*, *51*(5), 053018.



DIII-D Electron Cyclotron Emission Imaging (ECEi)

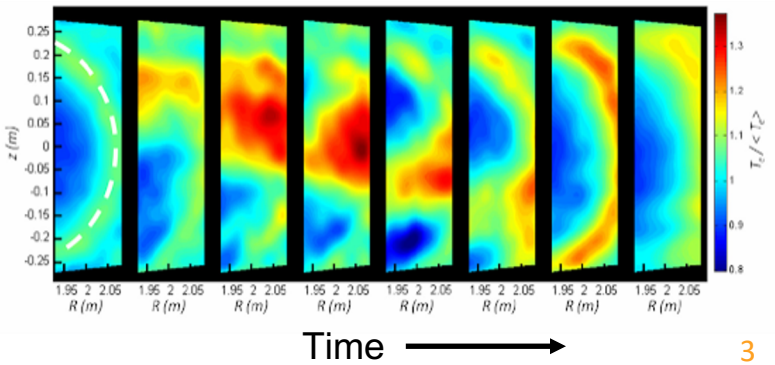
- Due to measurement of electron temperature, ECEi, is sensitive to a number of plasma phenomena important for disruptions, e.g.
  - Sawteeth
  - Tearing modes
  - ELM's
  - Impurity radiation (through drop in Te)
- Due to measurement being more local, can pick up on finer details of e.g. locked mode dynamics, poloidal mode numbers
- While combining multiple diagnostics for disruption predictions is desired, ECEi by itself has potential to capture pre-disruption events, on fast time scales
- ECEi characteristics:
  - Time resolution (1 MHz)
  - 20 x 8 channels for spatial resolution

[B. Tobias et al., RSI (2010)]
[Y. Zhu, HTPD2020]



DIII-D ECE Imaging since 2010, upgrade 2019.

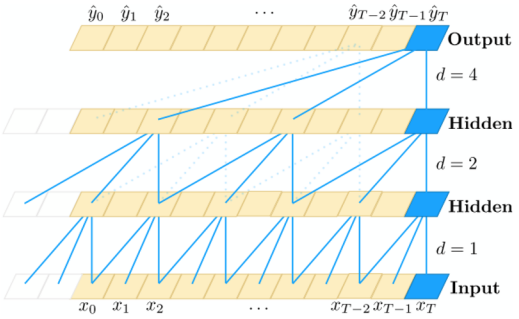
Zhu, Y., et al. Review of Scientific Instruments 91.9 (2020): 093504. [Editor Pick]

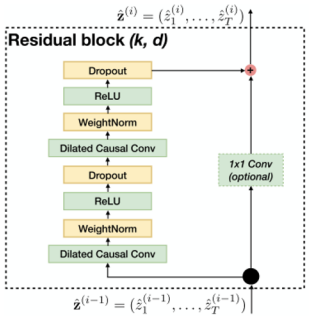


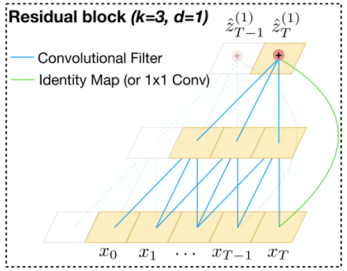
https://sites.google.com/view/mmwave/research/advanced-mmw-imaging/ecei-on-diii-d

Temporal Convolutional Networks

- Temporal Convolutional Network (TCN) architecture [*] combines causal, dilated convolutions with additional modern NN improvements (residual connections, weight normalization. etc.)
- Several beneficial aspects compared to RNN's:
  - Empirically TCN's exhibit longer memory (i.e. better for long sequences)
  - Non-sequential, allows parallelized training and inference
  - Require less GPU memory for training



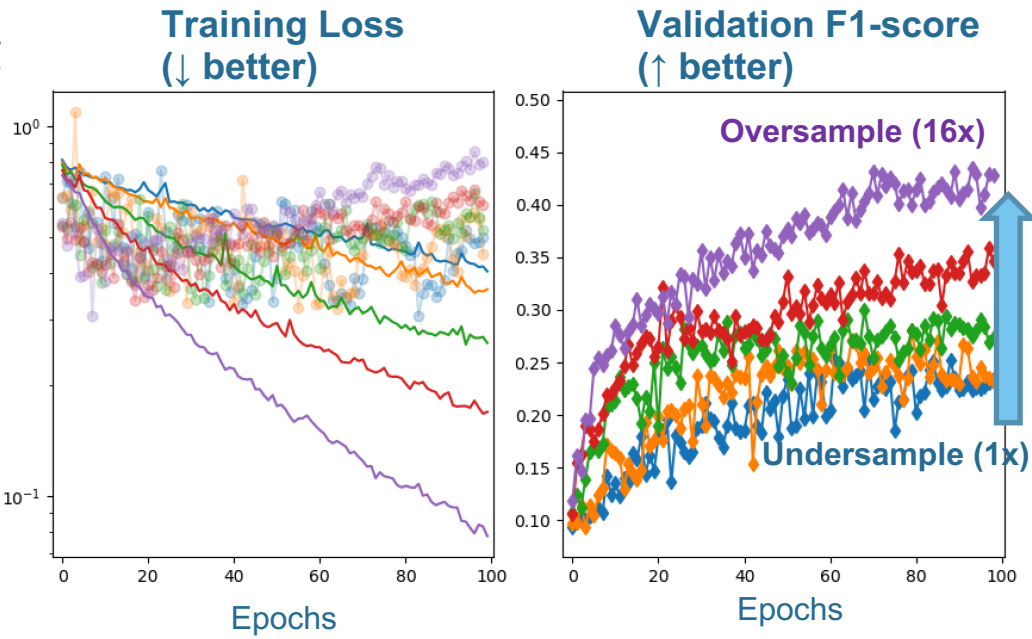




[* Bai, J.Z. Kolter, V. Koltun, http://arxiv.org/abs/1803.01271(2018)]

Using more data improves disruption prediction

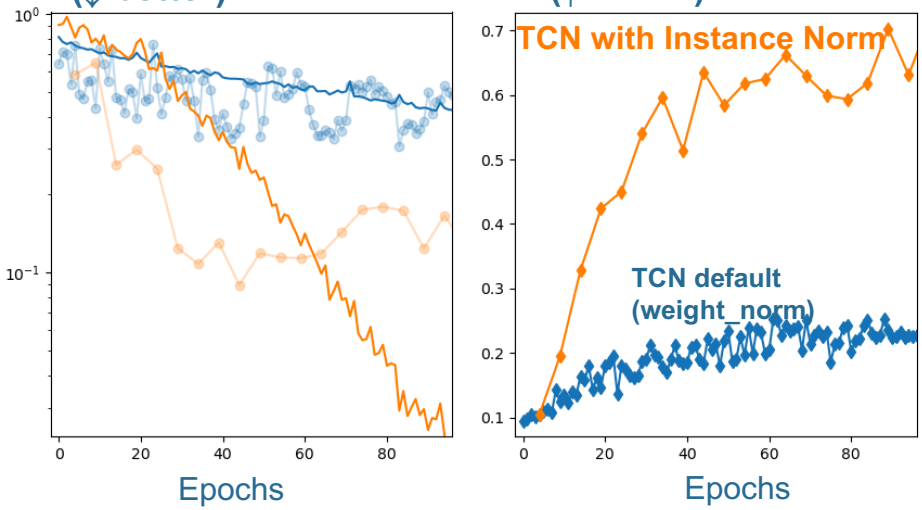
- Balancing datasets (disruptive sequences ~5% of dataset), utilizing as much non-disruptive data as possible gives best performance
  - Undersample (1x, ~60 GB) F1~0.22
    Churchill, R. M., et. al.(2020). *PoP*, *27*(6), 062510.
  - Oversample (16x, ~1 TB): F1~0.42
- Trained using 768 GPUs on Summit



Loss: measure of the "error"
F1-score: measure of correctly classified time-points
1 Epoch = 1 pass through ALL data

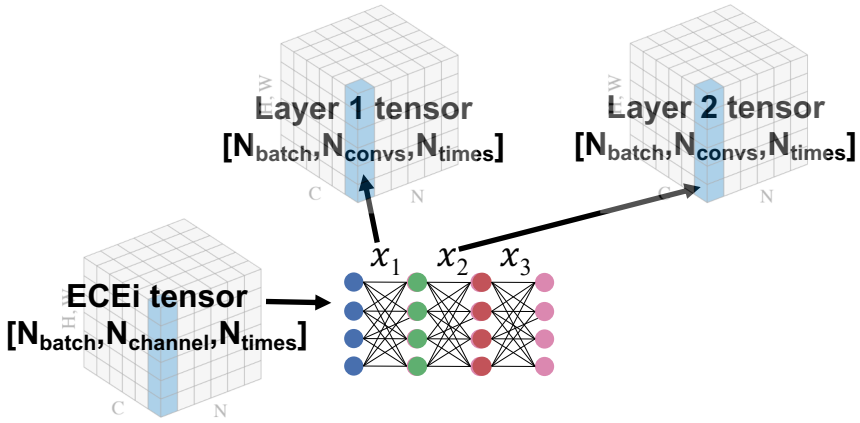
$$F_1 = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2}{\frac{TP+FP}{TP} + \frac{TP+FN}{TP}}$$

Feature Normalization prevents numerical instabilities in training



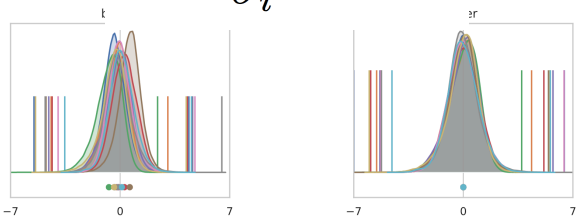
Serves to "damp out" numerical instabilities in NN training

https://arxiv.org/abs/1803.08494



Feature normalization normalizes the input to EACH layer across some dimension(s)

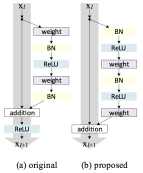
$$\hat{x}_i = \frac{1}{\sigma_i}(x_i - \mu_i)$$



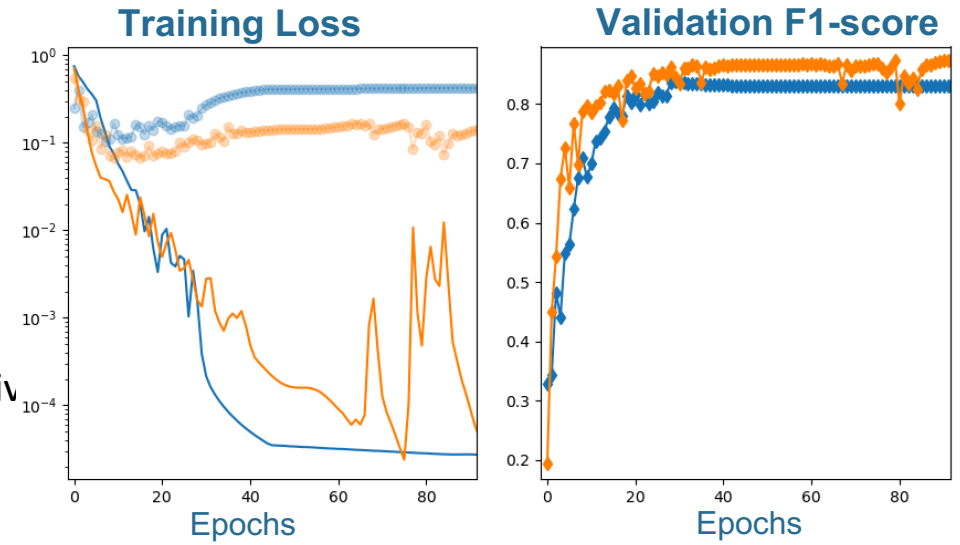
https://myrtle.ai/learn/how-to-train-your-resnet-7-batch-norm/

Neural network structure and optimizer give marginal improvements, but less sensitivity to hyperparameters

- Two “calibration” pieces gave marginal improvement, but much easier training (less time in tuning hyperparameters)



  - Placing feature normalization before convolutions removed need to clip gradients (avoid numerical instabilities)
  - Using AdamW optimizer much less sensitiv to learning rate (orders of magnitude range), presumably better for sequence models



Combined effect of more data(oversampling), instance normalization, and AdamW optimizer (+ others) lead to increase from F1~0.22 to F1~0.88 on time slice prediction

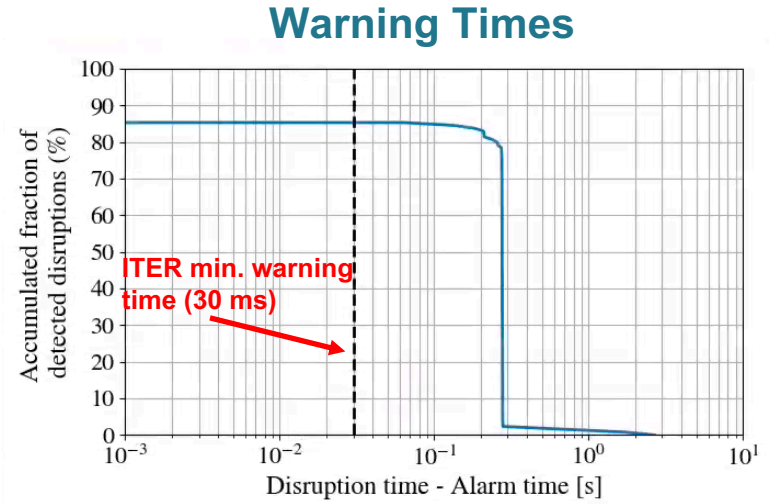
He, et. al. (2016) https://arxiv.org/pdf/1603.05027.pdf

Disruption prediction performance with ECEi data on DIII-D

- Shot performance using only ECEi data gives good disruption prediction (TPR~86% and FPR~7.5%), but not at level desired (TPR>95% and FPR<5%)
- Further improvements may come from:
  - Further improvements to NN architecture and/or training
  - incorporating additional diagnostics and physics

CONFUSION MATRIX FOR SHOT PREDICTIONS ON HOLDOUT TEST SET

		Predicted	
		Disruptive	Non-disruptive
Actual	Disruptive	TP=111	FN=18
	Non-disruptive	FP=12	TN=147



Summary

- A deep convolutional neural network with dilated convolutions (TCN) allows learning on multi-scale plasma diagnostics such as ECEi, for predicting complicated multi-physics phenomena such as tokamak disruptions
- Multiple improvements to existing TCN design such as Feature Normalization and oversampling minority classes give large performance gains
- The area of deep learning applied to plasma diagnostics has numerous potential applications for aiding fusion scientists

References

- AI quote: https://www.forbes.com/sites/peterhigh/2017/10/30/carnegie-mellon-dean-of-computer-science-on-the-future-of-ai/#3a6ad4f82197
- Gradient Descent, How Networks Learn https://www.youtube.com/watch?v=IHZwWFHWa
- HIT-II: A. J. Redd, B. A. Nelson, T. R. Jarboe, P. Gu, R. Raman, R. J. Smith, and K. J. McCollam, Physics of Plasmas 9, 2006 (2002)
- JET Shattered Pellet https://euro-fusion.org/news/2019/october/shattering-plasma-disruptions/
- Wendelstein 7-X https://www.youtube.com/watch?v=u-fbBRAxJNk
- Autodesk generative design https://www.youtube.com/watch?v=CtYRfMzmWFU
- GAN http://www.lherranz.org/2018/08/07/imagetranslation/
- Quotes from Alan Turing, Jeff Dean https://www.import.io/post/history-of-deep-learning/

References Cont.

- Meena, Google chatbot https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html
- Random forest for disruption prediction: C. Rea, R.S. Granetz, K. Montes, R.A. Tinguely, N. Eidietis, J.M. Hanson, B. Sammuli, Plasma Phys. Control. Fusion 60 (2018) 084004.
- SVM for disruption prediction: G.A. Rattá, J. Vega, A. Murari, S. Dormido-Canto, R. Moreno, Fusion Eng. Des. 112 (2016) 1014–1018.
- ML for control: Kolemen, Fu, Boyer, Erickson, to be published
- ML for simulation in control: Boyer
- ML for TGLF: O. Meneghini, et. al., Nucl. Fusion 57 (2017) 086034.
- https://myrtle.ai/learn/how-to-train-your-resnet-7-batch-norm/
- Mehta, V., et. al. (2020). http://arxiv.org/abs/2006.12682
- Churchill, R. M., et. al.(2020). PoP, 27(6), 062510. https://doi.org/10.1063/1.5144458

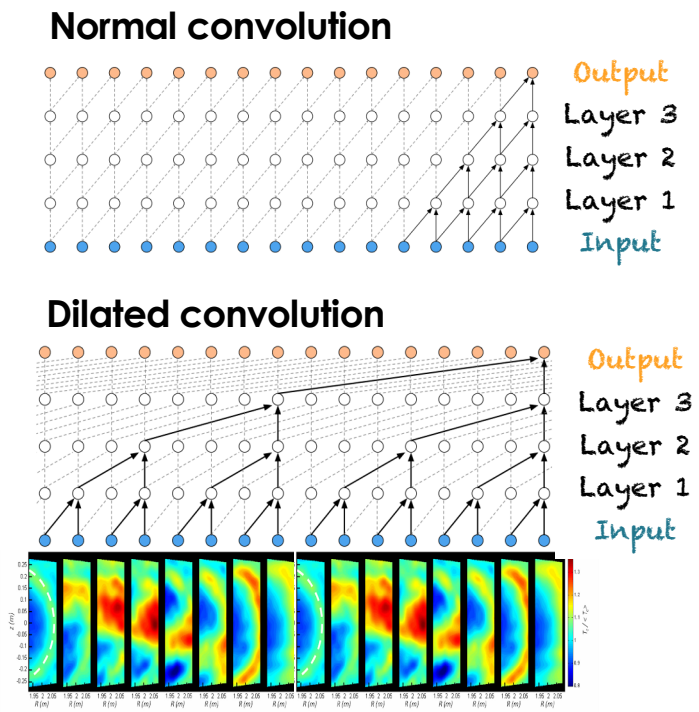
References Cont.

- https://twitter.com/MichaelAuli/status/1320755019432427520

Backup

Dilated convolutions enable efficient training on long sequences

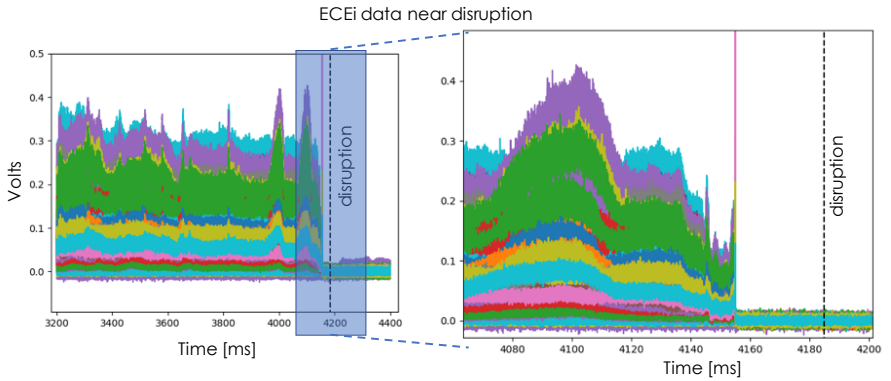
- Typical sequence neural networks such as Recurrent architectures (RNN, LSTM) have difficulties "remembering" long sequences of events [Bai 2018]
  - LSTM Rule of thumb sequence length <1000, so $\frac{T_{long}}{T_{short}} \lesssim 1000$
  - For disruptions with ECEi, $\frac{300\text{ ms}}{(100\text{ kHz})^{-1}}$ ~30,000
- CNNs with causal filters require large filters or many layers to learn from long sequences
  - Due to memory constraints, this becomes infeasible
- Dilated convolutions (i.e. convolution w/ defined gaps) increase the NN receptive field with same parameter size, allows training on high-time resolution diagnostic time series



[* A. Van Den Oord, et. al., WaveNET: A Generative Model for Raw Audio, 2016]

Dataset and computation

- Database of ~3000 shots (~50/50 non-disruptive/disruptive) with good ECEi data created from the Omfit DISRUPTIONS module shot list [E. Kolemen, et. al.]
  - "Good" data defined as all channels have SNR>3, avoid discharges where 2nd harmonic ECE cutoff
- ECEi data (~10 TB) transferred to various HPC centers for distributed training
  - Princeton U. TigerGPU (320 nVidia P100 GPUs)
  - PPPL/Princeton U. Traverse (186 nVidia V100 GPUs)
  - ORNL Summit (27,468 nVidia V100 GPUs)



Target setup for training neural network

- Target is to predict whether a time point is disruptive or not (binary classification)
- Time slices labelled “disruptive” 300 ms before disruption
  - Times before 350ms have similar distribution to non-disruptive discharges [Rea *FST* 2018]
  - Key assumption is that 300 ms before a disruption is a *minimum* amount of time by which events relating to disruptions will appear
  - When making shot predictions with NN timeslice predictions, will relax
- Each timeslice prediction uses previous 300ms
- Overlapping subsequences of length >> receptive field are created, length mainly set by GPU memory constraints

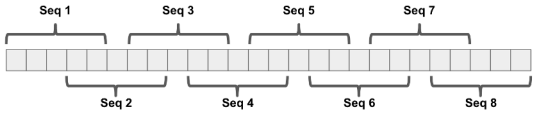
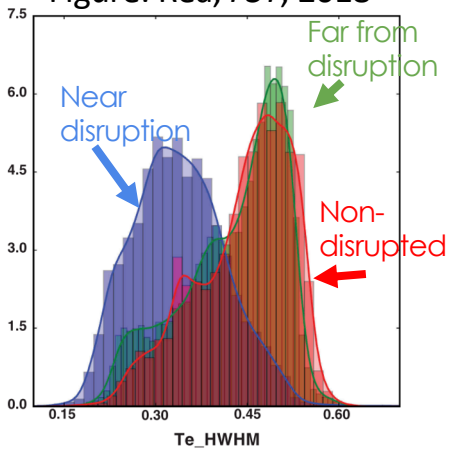
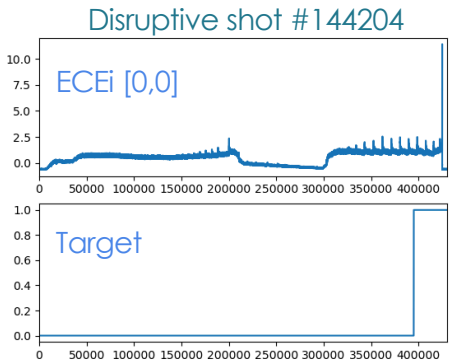


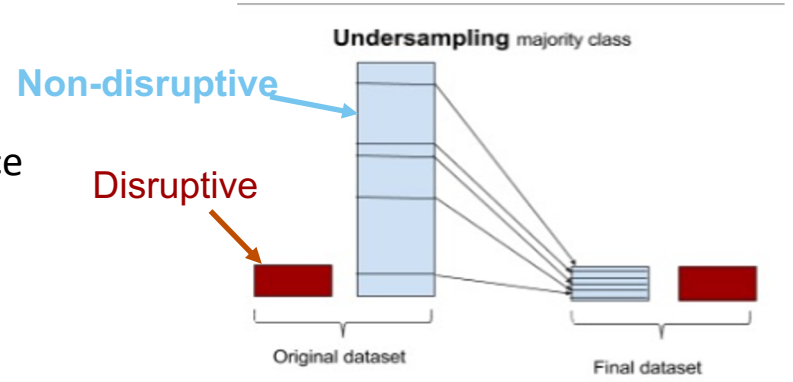
Figure: Rea, *FST*, 2018

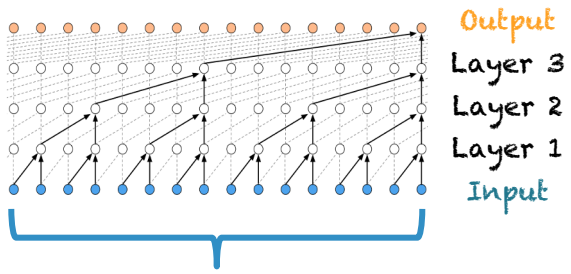




Data and network setup for training neural network

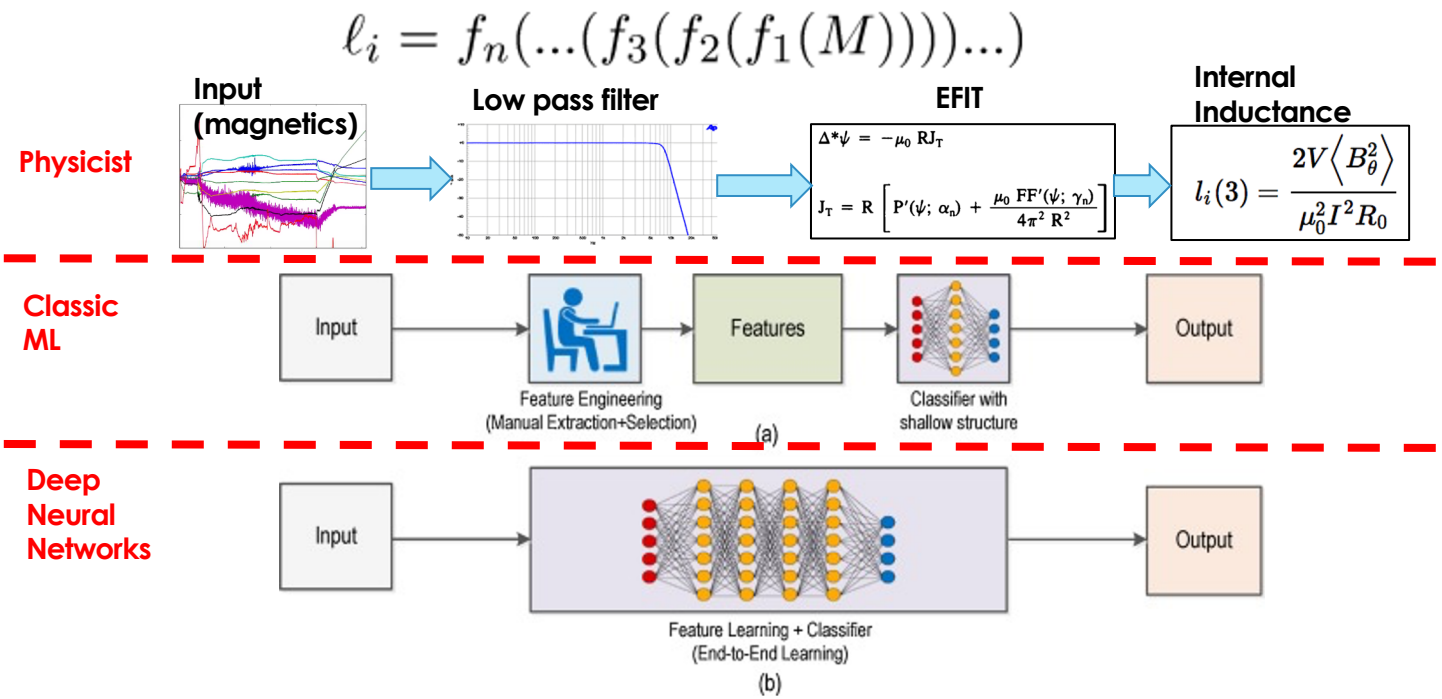
- Plan to start with smaller subsets of data
  - Downsample in time to 100 kHz
  - Undersample non-disruptive examples to balance dataset (natural class imbalance ~5% disruptive sequences)
- Neural network (TCN) setup:
  - Receptive field ~30,000 i.e. 300ms (each time slice prediction based on receptive field)
  - 4 layers, dilation 10, kernel size 15, hidden nodes 400 per layer





Receptive field 30,000
(# inputs needed to
make 1 prediction)

Deep learning enables working with complex, high-dimensional data

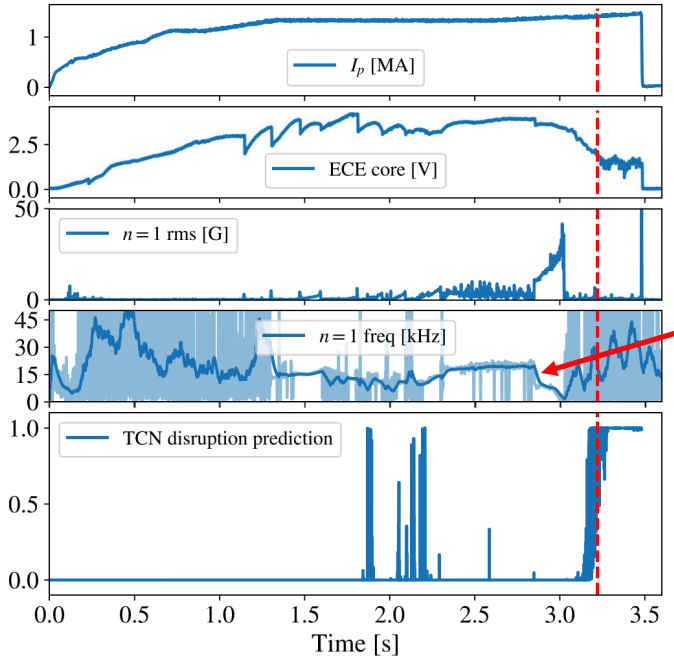


Hysteresis threshold method for shot predictions

- Performance on each time step is important, but predictions for each shot more common and useful to determine performance
- Since predictions done at each time step at 100 kHz, can have noisy prediction spikes
- We want mitigation to trigger when NN disruption predictor stays on long enough
  - Use hysteresis threshold method [Montes 2019] to determine alarm thresholds
  - Use Bayesian Optimization (optuna) to solve:

$$\boldsymbol{\theta}_* = \text{argmin}_{\boldsymbol{\theta}} \sqrt{[TPR(\mathbf{y};\boldsymbol{\theta}) - 1]^2 + [FPR(\mathbf{y};\boldsymbol{\theta}) - 0]^2}$$

$$\boldsymbol{\theta}_* = (\sigma_{low}, \sigma_{high}, \tau_{alarm})$$



Churchill, R. M., et. al.(2020). *PoP*, *27*(6), 062510.