- Ensemble decision trees perform better due to more interpretability, fast to train etc. compared to DNNs which are over parameterized and lack inductive bias.
- Still, DNNs have some pros like using multiple data types at once, continual learning, no need for feature engineering, end-to-end training etc.
- TabNet uses raw tabular data with gradient descent. It uses sequential attention for instance wise feature selection and uses single NN. It improves interpretability. It is the first network to use self-supervised learning for tabular data.
- It uses masked self-supervised labels to train encoder and decoder, then supervised training to train classifier based on encodings.
- Instance wise feature selection maximises mutual information between selected features and target variable.Categorical features are mapped to trainable embeddings.
- Tabnet uses soft feature selection with controllable sparsity and also output mapping in one network.
- It uses learnable mask (R^{BxD}) for soft feature selection to increase parameter efficiency.
- M[i] = sparsemax(P[i − 1] · hi(a[i − 1])). (sparsemax for each row/instance) h_i is trainable parameter. Prior scale $P[i] = \prod_{j=1}^{i} (\gamma - M[j])$. γ is relaxation param, as it increases from 1 to inf flexibility of using a feature in multiple decision steps increases. $P[0] = 1^{BxD}$. Some entries can be 0 in P[0] for self-supervised training.
- Sparsemax normalisation is superior in performance and produces good features.
- Sparsity regularisation is added as entropy to the final loss with a factor λ to further control sparsity.
- After feature transformer, resulting features are split for next step and predicting the output as [d[i], a[i]] = fi(M[i]·f) where $d[i] \in R^{BxNd}$ and $a[i] \in R^{BxNa}$. d[i] are all aggregated after applying ReLU to form final feature vector, and a[i] is used for transformer.
- Feature transformer consists of 2 shared layers (for robustness as same input features are used for all steps) and 2 step dependent layers. Skip connections are added with normalisation by sqrt(0.5) to control variance.
- Decoder takes encoding and aggregates output at each step to get final tabular features (estimate). It consists of feature transformer and an FC.
- Self-supervision : Reconstruction loss is L2 with normalisation by std of each feature, which is optimised to train decoder and also encoder. Masked features are decided by bernoulli distribution independently.
- Tabnet gave similar performance for global feature selection, but improved with instance-wise feature selection in which some indicators decided the salient features.
- $\eta_b[i] = \Sigma_c \, ReLU(d_{b,c}[i])$ is used to denote relative importance of i^{th} step for b^{th} example. The aggregated mask is also normalised across all features to make $\Sigma_j = 1$.
- N_d,N_a should not be too high (optimization problems), decreasing them does not degrade much. GLU is beneficial compared to ReLU.
- N_{steps} = [3,10]; $N_d = N_a$; larger N_{steps} => larger γ;