ON THE CONNECTION BETWEEN FISHER'S CRITERION AND SHANNON'S CAPACITY: THEORETICAL CONCEPTS AND IMPLEMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Fisher's criterion is arguably among the most widely used tools in machine learning for feature selection. The higher the value of Fisher's criterion, the more favorable a feature is. A rather different but nevertheless very important tool is Shannon's channel capacity. With Shannon's capacity, one can determine the maximum rate at which information can flow across a channel. Fisher's criterion and Shannon's capacity appear to be unrelated, yet both capture in their unique way the separation between probability distributions. In this study, we investigate whether Fisher's class separation criterion and Shannon's capacity can be related to each other. We formulate our research problem as a binary classification task and derive analytic expressions to determine if there is a potential link between Fisher's criterion and Shannon's capacity. It turns out that Fisher's class separation criterion and Shannon's channel capacity are intimately connected through two principal assumptions. Using this result, we develop a divergence measure for feature selection. Additionally, we show how our results can be used to solve classification problems and conduct a proof-of-concept experiment to demonstrate the viability of our approach.

1 INTRODUCTION

Feature selection is vital to the performance of a machine learning model. For a classification system, it identifies key features needed to distinguish between classes and discards those that can adversely impact performance. With fewer features, a model is simplified and its training speed is likely to improve. Knowing which features to keep, however, is a challenging task. A dataset with n features has 2^n possible combinations of features. As n increases, the search space quickly becomes prohibitively large. In this vast space of combinations, a feature selection scheme is required to find a subset of suitable features.

Fisher's criterion provides a scalable solution to select features. In many applications, Fisher's criterion is used to rank features by measuring their discriminative power. Once features are ranked, the best performing among them are selected for a model. Throughout the years, Fisher's criterion (Fisher, 1936) has been adopted by a wide variety of disciplines; its use nowadays can be found in diverse applications ranging from traffic-sign recognition (Zaklouta & Stanciulescu, 2012) to gene selection (Peng et al., 2006).

The Shannon capacity of an additive white Gaussian noise channel is perhaps the most recognizable result in information theory (Shannon, 1948). The Shannon capacity is a fundamental limit on the maximum amount of information that can flow across a channel. Research on the channel capacity has been mostly limited to the analysis of communication systems; very little is currently known about its potential for feature selection.

Fisher's criterion and Shannon's capacity appear to be unrelated, yet they both capture the separation between distributions in their unique way: Fisher's criterion captures this separation by gauging the ratio of the between-class variance to the within-class variance of probability distributions (Bishop, 2006), whereas the Shannon capacity captures the separation by measuring how far distributions are from independence (Appendix B). Knowing the relation between Fisher's and Shannon's measures, can provide insights for the design of feature selection schemes.

Contributions. The main contributions of this work are threefold:

- We show that Fisher's class separation criterion is related to the Shannon capacity of an additive white Gaussian noise channel (Section 3.1).
- Using the above result, we develop a novel divergence measure of class separation, which we used for feature selection (Section 3.2).
- We show how our theoretical results can be used to solve classification problems (Section 4).

2 Related Work

Feature selection is a process of selecting m features from a set n features ($m < n$) to enhance a model's performance. The source from which features are selected can be an original dataset. Alternatively, new features can be created from an original dataset via a mapping function; then, a feature selection scheme is subsequently employed to select a subset of features from this newly-created set. The process of creating new features from original ones is referred to as feature construction. Feature construction followed by feature selection is termed feature extraction (Guyon et al., 2008).

Based on the method employed to select a feature, the literature on feature selection can be organized into three categories: filters, wrappers, and embedded methods (Xue et al., 2015; Li et al., 2017). Here we will focus on filter methods as our results complement studies in this area.

In filter methods, features are ranked according to a given criterion. This approach is called a filter because it is used to "*filters out*" features that have low predictive power (Sebban & Nock, 2002). A fairly large number of filter methods have been proposed (Guyon et al., 2008; Bolón-Canedo et al., 2015); examples include:

Fisher's criterion. This criterion is a ratio of the between-class $(\mu_1 - \mu_0)^2$ to the within-class variance $(\sigma_1^2 + \sigma_0^2)$, where μ_k and σ_k^2 denote the mean and variance of a distribution, respectively. This ratio is used in Fisher's discriminant analysis, in which high-dimensional data is projected onto a line. The goal of this projection is to find a line on which Fisher's ratio is maximized. After which, a linear classifier is used to distinguish between classes (Fisher, 1936; Bishop, 2006). Fisher's ratio/criterion assesses the separation between class distributions with which features are ranked. The higher the value, the better a feature is for classification.

Volume of overlap. For a given feature, i, this quantity measures the amount of overlap between the tails of two class-conditional distributions and is defined as (Ho & Basu, 2002)

$$R_i = \frac{\min\{U_0^i, U_1^i\} - \max\{L_0^i, L_1^i\}}{\max\{U_0^i, U_1^i\} - \min\{L_0^i, L_1^i\}}\,, \tag{1}$$

here U_j^i and L_j^i denote the maximum and minimum value of each feature in class C_j, respectively. The lower the value of R_i, the higher a feature is ranked.

Pearson's correlation coefficient. This approach is a linear correlation measure. In this method, features and class labels are treated as random variables. The strength of the correlation between these two random variables is used to rank features (Arai et al., 2018).

Information-theoretic distances. Similar to Pearson's correlation coefficient approach, features and class labels are treated as random variables in information-theoretic measures (Duch et al., 2004). The key difference, however, between these two approaches is that information-theoretic distances can capture nonlinear dependencies between random variables.

The Kolmogorov–Smirnov test is a hypothesis test to determine whether samples of two classes are generated by the same distribution (Pratt & Gibbons, 1981). The lower the probability of the null hypothesis, the more likely a feature is beneficial for classification.

Relief method. In this approach, a sample x is randomly selected, without replacement, from a training set. Then, two distances, for a given feature, are measured: **1)** $d(x, x_s)$: the distance of x to its nearest neighbor x_s of the same class **2)** $d(x, x_d)$: the distance of x to its nearest neighbor

x_d of a different class. This process is repeated n times to obtain a relevance index J_i given by $J_i = J_{i\text{-}1} + \frac{1}{n}\Big(d(x, x_d) - d(x, x_s)\Big),\ i = 1, 2, \ldots, n$. A large value of J_i indicates that a feature has high relevance for classification (Kira & Rendell, 1992a;b).

Related work discussion. We position our work as a feature selection scheme. What sets our work apart from prior research efforts is that we develop a filter method (Section 3.2) by relating Fisher's criterion to Shannon's capacity. Research to date has not yet established this relationship.

3 Theoretical Results

3.1 Main result

This subsection details the connection between Fisher's class separation criterion and Shannon's channel capacity. We begin by listing our assumptions:

Assumptions:

- Gaussian distributions.

$$P(x \,|\, C_k) = \mathcal{N}(x\,;\, \mu_k,\, \sigma_k^2), \quad k \in \{0, 1\}$$

- A plausible assumption for a reliable classification system is that

$$\mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_i)P(C_i)\Big] \;\; > \mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_j)P(C_j)\Big], \qquad \forall\, i \neq j \in \{0, 1\}\,; \tag{2}$$

here $P(x|C_k)$ is a class-conditional distribution, and $P(C_k)$ is a class-prior probability.

Result: The mathematical expectation of $P(x|C_j)P(C_j)$ w.r.t. $P(x|C_i)$ is (see Appendix C for a detailed discussion)

$$\mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_j)P(C_j)\Big] = \frac{P(C_j)}{\sqrt{2\pi(\sigma_i^2 + \sigma_j^2)}}\; e^{-\frac{(\mu_i - \mu_j)^2}{2(\sigma_i^2 + \sigma_j^2)}}, \quad \text{where } \; i \neq j \in \{0, 1\}\,, \tag{3}$$

while for $P(x|C_i)P(C_i)$ the mathematical expectation is

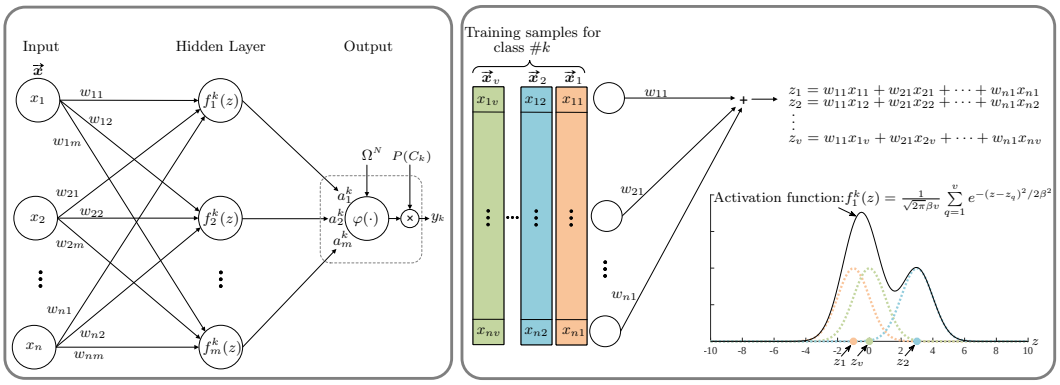
$$\mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_i)P(C_i)\Big] = \frac{P(C_i)}{2\sqrt{\pi}\sigma_i}, \quad \text{where } \; i \in \{0, 1\}\,. \tag{4}$$

Substituting Eqs. 3 and 4 into inequality 2; then taking the logarithm and rearranging, yields (see Appendix D for more details)

$$\mathcal{F} > 2\max\left\{\log\left(\frac{\sqrt{2}P(C_0)}{P(C_1)}\right) - \tilde{\mathcal{C}}_0 \;\;,\;\; \log\left(\frac{\sqrt{2}P(C_1)}{P(C_0)}\right) - \tilde{\mathcal{C}}_1\right\}, \tag{5}$$

here $\mathcal{F} := \frac{(\mu_1 - \mu_0)^2}{\sigma_1^2 + \sigma_0^2}$ is Fisher's criterion (Bishop, 2006) and $\tilde{\mathcal{C}}_k := \frac{1}{2}\log\left(1 + \frac{\sigma_k^2}{\sigma_{1\text{-}k}^2}\right)$ is the Shannon capacity of an additive white Gaussian noise channel[1] (Shannon, 1948; Cover & Thomas, 2006).

[1]The unit of $\tilde{\mathcal{C}}_k$ here is *nats per real dimension* (1 nat $\approx$ 1.44 bits).



(a) (b)

Figure 1: (a) Architecture of proposed scheme (b) Construction of activation functions.

3.2 Proposed divergence measure

Using inequality 5, we suggest the following divergence measure to capture the separation between class distributions:

$$\tilde{\mathcal{D}} = \mathcal{F} - \mathcal{T}\,, \tag{6}$$

where $\mathcal{T}$ is the RHS of inequality 5:

$$\mathcal{T} = 2\max\left\{\log\left(\frac{\sqrt{2}P(C_0)}{P(C_1)}\right) - \tilde{\mathcal{C}}_0\ ,\ \log\left(\frac{\sqrt{2}P(C_1)}{P(C_0)}\right) - \tilde{\mathcal{C}}_1\right\}. \tag{7}$$

$\tilde{\mathcal{D}}$ enables one to measure the discriminative power of a given feature. The higher the value of $\tilde{\mathcal{D}}$, the more discriminative power a feature holds. In Section 5, we will employ $\tilde{\mathcal{D}}$ in our experiments to help select a subset of features used for image classification.

4 Model

The goal of this section is to illustrate how our theoretical results in Section 3 can be used to solve classification problems. To this end, we propose a neural network architecture followed by a description of its activation functions. In Section 5, we will test the performance of the proposed network.

4.1 Architecture and principles of operation

Figure 1a shows the architecture of the proposed neural network. The network consists of three layers: an input, hidden, and output layer. Let m denote the total number of nodes in the hidden layer. For each node in the hidden layer, an input vector $\vec{\boldsymbol{x}} \in \mathbb{R}^n$ is multiplied by random weights to form z, which is fed to an activation function $f_i^k(z)$. Here $f_i^k(z)$ is the i^{th} activation function for class k. Outputs of activation functions, represented by set $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$, are in turn fed to $\varphi(\cdot)$. Function $\varphi(\cdot)$ selects a subset, Ω^N (Section 5), of elements from $\mathcal{A}$ and sums them. This summation is computed for all classes ($k = 0, 1$) and multiplied by a corresponding class prior probability, $P(C_k)$, to obtain a set of outputs values $\mathcal{Y} = \{y_0, y_1\}$. The predicted class is the argument of set $\mathcal{Y}$ that yields the maximum value: $k^* = \underset{k\in\{0,1\}}{\operatorname{argmax}}\, y_k$. **Algorithm** 1 summarizes the proposed scheme, and an expanded illustration of the proposed scheme is provided in Fig. 7 (Appendix G).

Algorithm 1: Proposed Scheme

Input: $\vec{\boldsymbol{x}}, \{P(C_k)\}, \{f_i^k(z)\}, \Omega^N$ ▷ $\vec{\boldsymbol{x}} \in \mathbb{R}^n$: input vector of unknown class. $\{P(C_k)\}$: set of class-prior probabilities. $\{f_i^k(z)\}$: set of activation functions (computed via Algorithm 2). Ω^N: set of indices of divergence measurements (Section 5).

Output: k^* ▷ Here k^* is the predicted class of $\vec{\boldsymbol{x}}$.

$$W \leftarrow \begin{bmatrix} — & w_1^T & — \\ — & w_2^T & — \\ & \vdots & \\ — & w_m^T & — \end{bmatrix}_{m \times n}$$

▷ Generate random matrix W. In our experiment, the entries of W are independent and drawn from a standard normal distribution.

for $k = 0, 1$ **do** ▷ Here k is the index of a class, and m is the number of nodes in the hidden layer.

  for $i = 1, \ldots, m$ **do**

    $z_i \leftarrow w_i^T \vec{\boldsymbol{x}}$

    $a_i^k \leftarrow f_i^k(z_i)$

  end

  $y_k \leftarrow P(C_k) \sum_{j \in \Omega^N} a_j^k$ ▷ Summation of outputs, a_j^k, of N activation functions that have the largest divergence values(φ in Fig. 1a), followed by a multiplication by a class's prior probability.

end

$k^* \leftarrow \underset{k \in \{0,1\}}{\operatorname{argmax}}\, y_k$

4.2 ACTIVATION FUNCTIONS

The key idea here is to employ class-conditional probability density functions, obtained from training samples, as activation functions $\left(f_i^k(x)\right)$. In Section 3.1 we have $P(x|C_k)$, and a way to realize such density functions in practice is to represent them as activation functions—that is, we use $f_i^k(x)$ to mimic $P(x|C_k)$. Fig. 1b illustrates how an activation function is constructed. Each class, k, has a dedicated network (Fig. 1a). For each node in this network, training data $\{\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_2, \ldots, \vec{\boldsymbol{x}}_v\}$, of the same class, are multiplied by random weights to obtain a sample set $\{z_1, z_2, \ldots, z_v\}$. This set is used by a kernel density estimator (KDE) to construct an activation function as follows:

$$f_i^k(z) = \frac{1}{\sqrt{2\pi}\beta v} \sum_{q=1}^{v} e^{-(z-z_q)^2/2\beta^2}, \tag{8}$$

here subscript i indicates the i^{th} node of a given class k, while v and β denote the number of training samples and bandwidth of the KDE, respectively. A variety of methods can be used to find an apt value of β (see, for instance, Silverman (1986); Sheather & Jones (1991); Scott (1992), as well as Heidenreich et al. (2013) for a review). In our experiment, functions $f_i^k(z)$ are computed on-demand rather than saved as lookup tables; this approach is computationally heavy but saves a great amount of memory. The proposed neural network learns without weight tuning: learning is accomplished by letting data of training samples shape activation functions (Fig. 1b). **Algorithm** 2 provides a summary of how activation functions are constructed.

Algorithm 2: Construction of Activation Functions

Input: $\{\mathbf{x}_q^k\}_{k=0}^1$ ▷ Training set; $\mathbf{x}_q^k \in \mathbb{R}^n$, where k indicates a given class, and q is a sample's index.

Output: $f_i^k(z)$ ▷ Activation functions.

$$W \leftarrow \begin{bmatrix} — & w_1^T & — \\ — & w_2^T & — \\ & \vdots & \\ — & w_m^T & — \end{bmatrix}_{m \times n}$$

▷ Generate random matrix W. This matrix is identical to W of Algorithm 1.

for $k = 0, 1$ **do**
  for $i = 1, \ldots, m$ **do**
    for $q = 1, \ldots, v_k$ **do**
      $z_q \leftarrow w_i^T \mathbf{x}_q^k$
    end
    $\beta_i \leftarrow F(z_1, \ldots, z_{v_k})$
    $f_i^k(z) \leftarrow \frac{1}{\sqrt{2\pi}\beta_i v_k} \sum_{q=1}^{v_k} e^{-(z-z_q)^2/2\beta_i^2}$
  end
end

▷ Here m is the number of nodes in the hidden layer and v_k is the number of training samples for a given class k.

▷ Here β_i is the bandwidth of the kernel density estimator, determined by $F(\cdot)$. In our experiment, $F(\cdot) = \left(\frac{4}{3v_k}\right)^{\frac{1}{5}} \times \left(\frac{\eta}{0.6745}\right)$, where η is the median absolute deviation of set $\{z_1, \ldots, z_{v_k}\}$.

5 EXPERIMENT

Task description: Binary classification.

Dataset: Pairs of image classes are obtained from the Fashion-MNIST dataset (Xiao et al., 2017); this dataset was z-normalized.

Methods:

- **Kernel density estimator (KDE).** Gaussian functions $\left(K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}u^2}\right)$ are used as kernels for the KDE $\left(f(z) = \frac{1}{\beta v} \sum_{q=1}^{v} K\left(\frac{z-z_q}{\beta}\right)\right)$. The bandwidth of kernels are calculated as $\beta = \left(\frac{4}{3v}\right)^{\frac{1}{5}} \sigma$, where σ is a measure of dispersion (spread) and v is the number of samples of a set (Silverman, 1986). The standard deviation can be used to measure dispersion. In our experiment, however, we chose the median absolute deviation, η, as it provides a more robust dispersion measure when outliers are present $\left(\sigma = \frac{\eta}{0.6745}\right.$, Viertl (2009); Förstner & Wrobel (2016)$\left.\right)$.
- **Divergence calculations**. To deal with outliers when calculating the divergence in Eq. 6, we use the interquartile mean to approximate μ (Sprent, 2012), and $\sigma \approx \frac{\eta}{0.6745}$ to approximate the standard deviation. Priori probabilities, $P(C_k)$, are estimated as the proportion of each class in a training dataset.
- **Subset** Ω^N. Consider Fig. 8 (Appendix G). Ω^N is a subset of the nodes that have the largest N divergence values. Let $\vec{\mathcal{D}} = \{\tilde{\mathcal{D}}_1, \tilde{\mathcal{D}}_2, \ldots, \tilde{\mathcal{D}}_m\}$ denote a set of divergence measurements, where $\tilde{\mathcal{D}}_i$ denotes the divergence measured at the i^{th} node, and m is the total number of nodes in the hidden layer. The value of a given $\tilde{\mathcal{D}}_i$ is computed via Eq. 6. Additionally, let $\Omega = \{\omega_1, \ldots, \omega_N, \ldots, \omega_m\}$ denote the set of indices of the sorted values of $\vec{\mathcal{D}}$ in descending order. For example, if $\vec{\mathcal{D}} = \{\tilde{\mathcal{D}}_1, \tilde{\mathcal{D}}_2, \tilde{\mathcal{D}}_3\}$ and $\tilde{\mathcal{D}}_2 > \tilde{\mathcal{D}}_3 > \tilde{\mathcal{D}}_1$, then $\Omega = \{2, 3, 1\}$. Using Algorithm 1 with the training dataset, the value of N is determined by direct search: $N = 1, 2, \ldots m$. The value of N that yields the highest ac-

curacy is chosen (Fig. 2 provides an illustration). To keep the notation light, let subset $\Omega^N = \{\omega_1, \omega_2, \ldots, \omega_N\}$ denote the first N values in set Ω. For each class, 30% of the training set is kept for validation during the search for N.

- **Weights**. The traditional approach to train neural networks is weight tuning, whereby weights of a network are computed either iteratively or by solving a set of equations so as to minimize a discrepancy between a model's prediction and a corresponding ground truth (Broomhead & Lowe, 1988; Huang et al., 2006; Schmidhuber, 2015). In this study, we provide an alternative approach. We illustrate how neural networks can be designed to classify patterns without weight tuning: weights of networks herein are randomly generated and left afterward unchanged. The weights, w_{ij}, of the network (Fig. 1a) are drawn independently from a standard normal distribution.
- **Randomization.** We run our network with 10 different randomization seeds to report a 95% confidence interval. The randomization is in the weights of the network.

Experimental results: Figures 3 (b)–(e) show responses of activation functions, a_i^k, to unseen test images. By and large, activation functions have a relatively high response when they are of the same class as a test image; this is particularly noticeable in Fig. 3 (b), where activation functions of class $\#0$ predominantly produce responses larger than that of other classes. Moreover, Figs. 4 (a)–(d) show the classification accuracy of the proposed network versus the number of nodes in the hidden layer. What immediately stands out is that a high accuracy is achieved when the classification is between objects that are perceptually dissimilar and vice versa. Consider Fig. 4 (a) as an example; the classification between a T-shirt (class $\#0$) and sneaker (class $\#7$) has a higher accuracy than between a T-shirt (class $\#0$) and a formal shirt (class $\#6$). Additionally, the classification accuracy increases with the number of nodes in the network and then plateaus. This trend can be expected because the possibility of obtaining a subset of nodes, Ω^N, with high divergence values increases with the number of nodes available, albeit in a nonlinear manner. The interested reader is referred to Appendices E and F for a diverse range of comparisons.

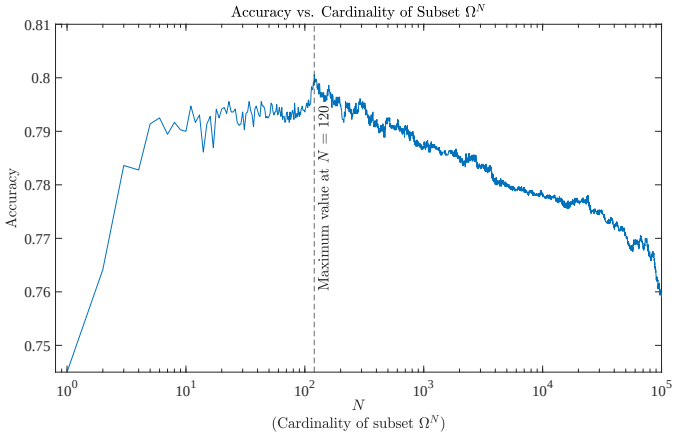
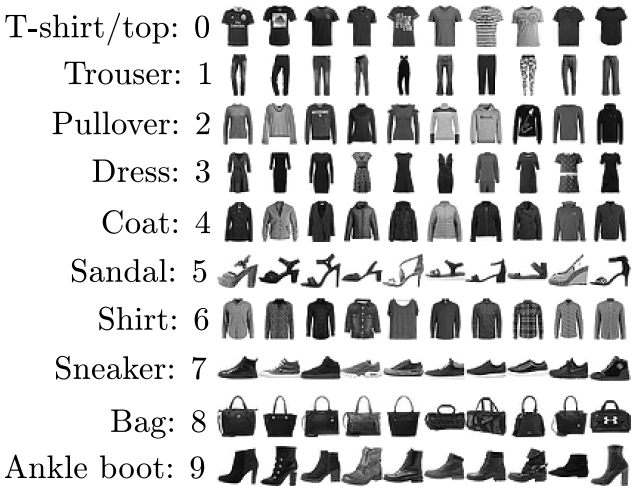


Figure 2: Accuracy versus cardinality, N, of subset Ω^N; carried out on the training dataset using Algorithm 1. The aim of this experiment is to determine the value of N, which is required by Algorithm 1 during the testing phase. In this example, the binary classification task is between class $\#2$ (pullover) and $\#4$ (coat). The highest accuracy is $\sim 80\%$ and occurs at a cardinality of size $N = 120$; this value of N is obtained by a brute-force search. The number of nodes in the hidden layer of the network is $m = 10^5$.



(a) Samples from dataset.

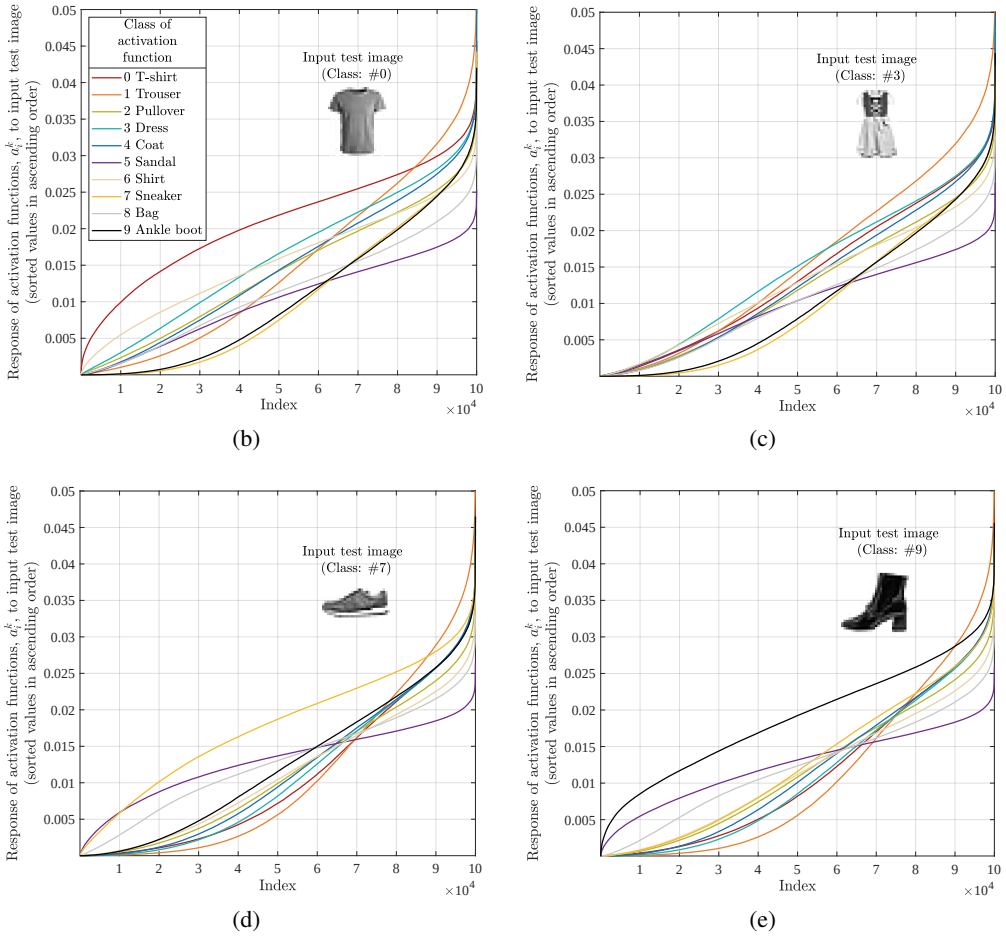


Figure 3: (b)–(e) Responses of activation functions, a_i^k, to unseen test images. The y-axis displays the sorted values of set $\{a_1^k, a_2^k, \ldots, a_m^k\}$ in ascending order, while x-axis is the index of said sorted values (see Fig. 1a for a schematic illustration depicting a_i^k responses). The number of nodes in the hidden layer of the network is $m = 10^5$.

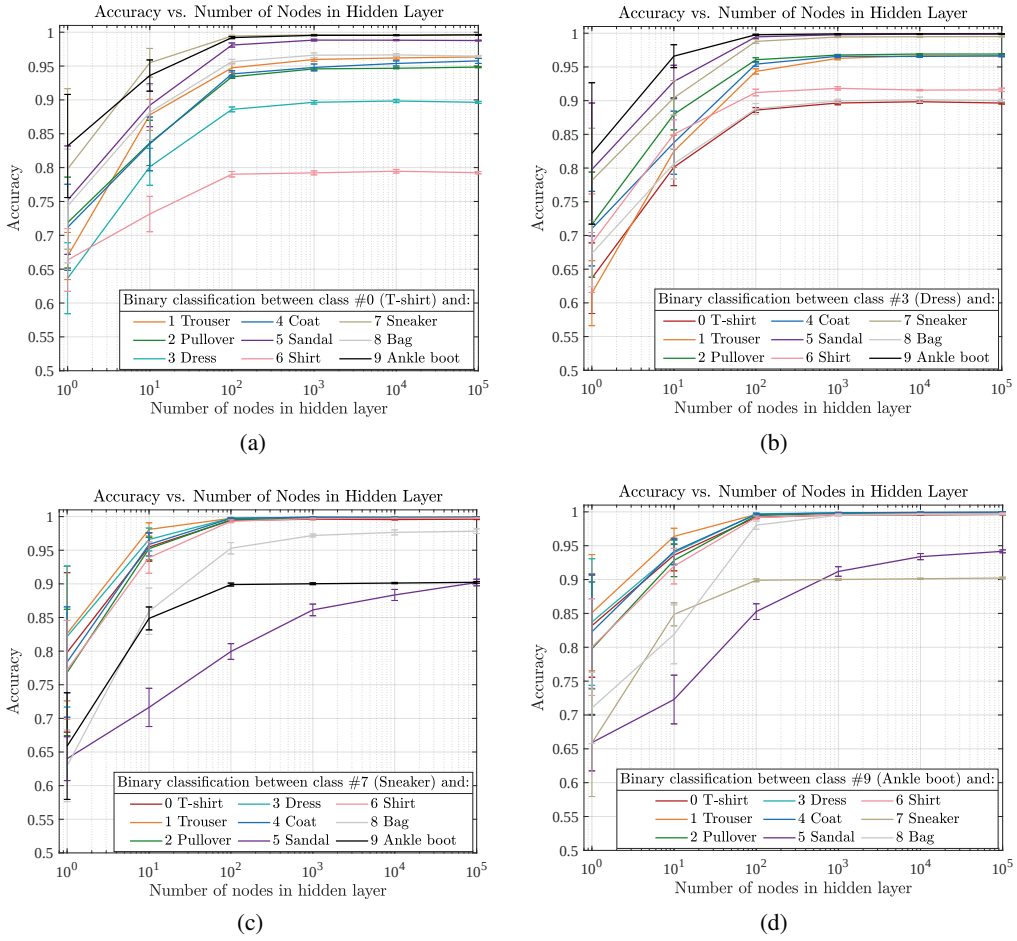


Figure 4: Classification accuracy versus number of nodes, m, in the hidden layer of the neural network. Error bars are 95% confidence intervals and are computed by randomizing weights of the network a total of 10 times.

6 Conclusion and Future Work

Summary. This paper set out to establish a link between Fisher's criterion and Shannon's channel capacity. It is shown that Fisher's class separation criterion and the Shannon capacity of an additive white Gaussian noise channel are related through two assumptions. Using this result, a divergence measure is developed and used as a filter method for a proposed neural network. Experimental results demonstrate that the techniques devised herein to solve classification problems hold potential.

Limitations and future work. The scope of this study was limited to a binary classification setting. As such, a natural progression of this work is to generalize our results to multiple classes. Additionally, the proposed divergence measure ranks features individually according to their discriminative power. While this approach has a low computational requirement, it neglects possible synergistic interactions among features. A promising research direction, therefore, is to explore the impact of said interactions on classification performance.

References

K. Arai, S. Kapoor, and R. Bhatia. *Intelligent Computing: Proceedings of the 2018 Computing Conference, Volume 1*. Advances in Intelligent Systems and Computing. Springer International Publishing, 2018. ISBN 9783030011741.

C.M. Bishop. *Pattern Recognition and Machine Learning*, pp. 188. Information Science and Statistics. Springer, 2006. ISBN 13 978-0387-31073-2.

Verónica Bolón-Canedo, Noelia Sánchez-Maroño, and Amparo Alonso-Betanzos. *Feature selection for high-dimensional data*. Springer, 2015. ISBN 978-3-319-21857-1.

Paul Bromiley. Products and convolutions of gaussian probability density functions. *Tina-Vision Memo*, 3(4):1, 2003.

D Broomhead and D Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2, 1988.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory 2nd edition*. John Wiley & Sons, Inc., New Jersey, USA, 2006. ISBN 13 978-0-471-24195-9.

Włodzisław Duch, Tadeusz Wieczorek, Jacek Biesiada, and Marcin Blachnik. Comparison of feature ranking methods based on information entropy. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, volume 2, pp. 1415–1419. IEEE, 2004.

Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7 (2):179–188, 1936.

Wolfgang Förstner and Bernhard P Wrobel. *Photogrammetric computer vision*, pp. 40. Springer, 2016. ISBN 978-3-319-11549-8.

Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207, pp. 1–4,5–637. Springer, 2008. ISBN 13-978-3-540-35487-1.

Nils-Bastian Heidenreich, Anja Schindler, and Stefan Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. *AStA Advances in Statistical Analysis*, 97(4):403–433, 2013.

Tin Kam Ho and Mitra Basu. Complexity measures of supervised classification problems. *IEEE transactions on pattern analysis and machine intelligence*, 24(3):289–300, 2002.

Guang-Bin Huang, Qin-Yu Zhu, and Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501, 2006.

Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pp. 129–134, 1992a.

Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine learning proceedings 1992*, pp. 249–256. Elsevier, 1992b.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

Yanxiong Peng, Wenyuan Li, and Ying Liu. A hybrid approach for biomarker discovery from microarray gene expression data for cancer classification. *Cancer informatics*, 2, 2006.

J.W. Pratt and J.D. Gibbons. *Concepts of Nonparametric Theory*. Springer Series in Statistics. Springer New York, 1981. ISBN 13:978-1-4612-5933-6.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

D. W. Scott. *Density Estimation for Statistics and Data Analysis*. John Wiley & Sons, Inc., 1992. ISBN 9780470316849.

Marc Sebban and Richard Nock. A hybrid filter/wrapper approach of feature selection using information theory. *Pattern recognition*, 35(4):835–846, 2002.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53 (3):683–690, 1991.

B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. Taylor & Francis, 1986. ISBN 9780412246203.

P. Sprent. *Applied Nonparametric Statistical Methods*. Springer Netherlands, 2012. ISBN 9789400912236.

R. Viertl. *PROBABILITY AND STATISTICS - Volume II: Probabilistic Models and Methods Foundations of Statistics*, pp. 220. Encyclopedia of Life Support Systems; Mathematical Sciences. EOLSS Publishers Company Limited, 2009. ISBN 9781848260535.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv:1708.07747*, 2017.

Bing Xue, Mengjie Zhang, Will N Browne, and Xin Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2015.

Fatin Zaklouta and Bogdan Stanciulescu. Real-time traffic-sign recognition using tree classifiers. *IEEE Transactions on Intelligent Transportation Systems*, 13(4):1507–1514, 2012.

APPENDIX

A NOTATION

Numbers and Arrays

a	A scalar (integer or real)
$\vec{\boldsymbol{x}}$	A vector
W	A matrix
w_i^T	Row i of matrix W

Sets

$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1

Probability

$P(C_k)$	Class-prior probability
$P(x\|C_k)$	Class-conditional distribution
$X \sim P$	Random variable X has distribution P
$\mathop{\mathbb{E}}_{X\sim P(x)}[f(x)]$	Mathematical expectation of $f(x)$ with respect to $P(x)$
$\text{Var}[X]$	Variance of random variable X
$\mathcal{N}(x; \mu, \sigma^2)$	Gaussian distribution over x with mean μ and variance σ^2

Functions

$\log x$	The natural logarithm of x
$\min\{x, y\}$	The minimum of the two numbers
$\max\{x, y\}$	The maximum of the two numbers
$\underset{i}{\text{argmax}}\, y_i$	The argument of the maxima
$D_{\text{KL}}\Big(P \,\|\|\, Q\Big)$	The Kullback-Leibler divergence of P from Q

B SEPARATION: DISTANCE FROM INDEPENDENCE

Consider the following example in which we have three random variables: Y, X_k and $X_{1\text{-}k}$, related by

$$Y = \underbrace{X_k}_{\text{signal}} + \underbrace{X_{1\text{-}k}}_{\text{noise}}\,, \qquad \text{here } k \in \{0, 1\} \text{ is a label of a class.} \tag{9}$$

The Shannon capacity, $\tilde{\mathcal{C}_k}$, captures how far the joint probability density function of X_k and Y $\Big(\text{i.e., } P(x, y)\Big)$ is from independence $\Big(\text{i.e., } P(x)P(y)\Big)$ through the maximization of the Kullback-Leibler divergence:

$$\tilde{\mathcal{C}_k} = \max_{P(x):\ \text{Var}[X_k]\leq\sigma_k^2} D_{\text{KL}}\Big(P(x, y)\Big\|P(x)P(y)\Big)\,. \tag{10}$$

For Gaussian noise: $X_{1\text{-}k} \sim P(x \,|\, C_{1\text{-}k}) = \mathcal{N}(x\,;\, \mu_{1\text{-}k},\, \sigma_{1\text{-}k}^2)$, Eq. 10 reduces to

$$\tilde{\mathcal{C}_k} = \frac{1}{2}\log\left(1 + \frac{\sigma_k^2}{\sigma_{1\text{-}k}^2}\right) \;\; \textit{nats per real dimension}, \qquad k \in \{0, 1\}, \tag{11}$$

which is attained when $X_k \sim P(x \,|\, C_k) = \mathcal{N}(x\,;\, \mu_k,\, \sigma_k^2)$; (see Shannon (1948); Cover & Thomas (2006) for a comprehensive discussion).

C MATHEMATICAL EXPECTATIONS

Here we expand on our discussion in Section 3.1 on Eqs. 3 and 4. The mathematical expectation of $P(x|C_j)P(C_j)$ w.r.t. $P(x|C_i)$ is

$$\mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_j)P(C_j)\Big] = P(C_j)\int_{-\infty}^{\infty} P(x|C_j)P(x|C_i)\,dx\;;\;\text{where}\;\; i\neq j\in\{0,1\}\;\;, \tag{12}$$

here we have a product of two Gaussian functions, $P(x|C_j)P(x|C_i)$; this product is a Gaussian function with:

$$\text{a mean: } \tilde{\mu} = \frac{\mu_i\sigma_j^2+\mu_j\sigma_i^2}{\sigma_i^2+\sigma_j^2}\;\;, \tag{13}$$

$$\text{standard deviation: } \tilde{\sigma} = \frac{\sigma_i\sigma_j}{\sqrt{\sigma_i^2+\sigma_j^2}}\;\;, \tag{14}$$

$$\text{and scaling factor of: } \tilde{S} = \frac{1}{\sqrt{2\pi(\sigma_i^2+\sigma_j^2)}}\;\; e^{-\frac{(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}} \tag{15}$$

(see Bromiley (2003) for a detailed discussion). Using Eqs. 13–15 in 12, it follows that

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_j)P(C_j)\Big] &= P(C_j)\tilde{S}\int_{-\infty}^{\infty}\mathcal{N}\left(x\,;\,\tilde{\mu}\,,\tilde{\sigma}^2\right)\,dx\\ &= P(C_j)\tilde{S}\\ &= \frac{P(C_j)}{\sqrt{2\pi(\sigma_i^2+\sigma_j^2)}}\;\; e^{-\frac{(\mu_i-\mu_j)^2}{2(\sigma_i^2+\sigma_j^2)}}\;\;.\end{aligned} \tag{16}$$

Likewise,

$$\begin{aligned}\mathop{\mathbb{E}}_{X\sim P(x|C_i)}\Big[P(x|C_i)P(C_i)\Big] &= P(C_i)\int_{-\infty}^{\infty}P(x|C_i)P(x|C_i)\,dx\\ &= \frac{P(C_i)}{2\sqrt{\pi}\sigma_i}\;\;.\end{aligned} \tag{17}$$

D SUPPORTING DETAILS FOR SECTION 3.1

Here we add some supporting details to our discussion in Section 3.1. Consider a binary classification problem in which we have C_0 and C_1. For this setting, we need two conditions to be simultaneously satisfied, namely:

Condition 1:

$$\mathop{\mathbb{E}}_{X\sim P(x|C_0)}\Big[P(x|C_0)P(C_0)\Big] \;>\; \mathop{\mathbb{E}}_{X\sim P(x|C_0)}\Big[P(x|C_1)P(C_1)\Big]$$

$$\frac{(\mu_1-\mu_0)^2}{\sigma_1^2+\sigma_0^2} \;>\; 2\log\left(\frac{\sqrt{2}P(C_1)}{P(C_0)}\right) - \log\left(1+\frac{\sigma_1^2}{\sigma_0^2}\right), \tag{18}$$

and

Condition 2:

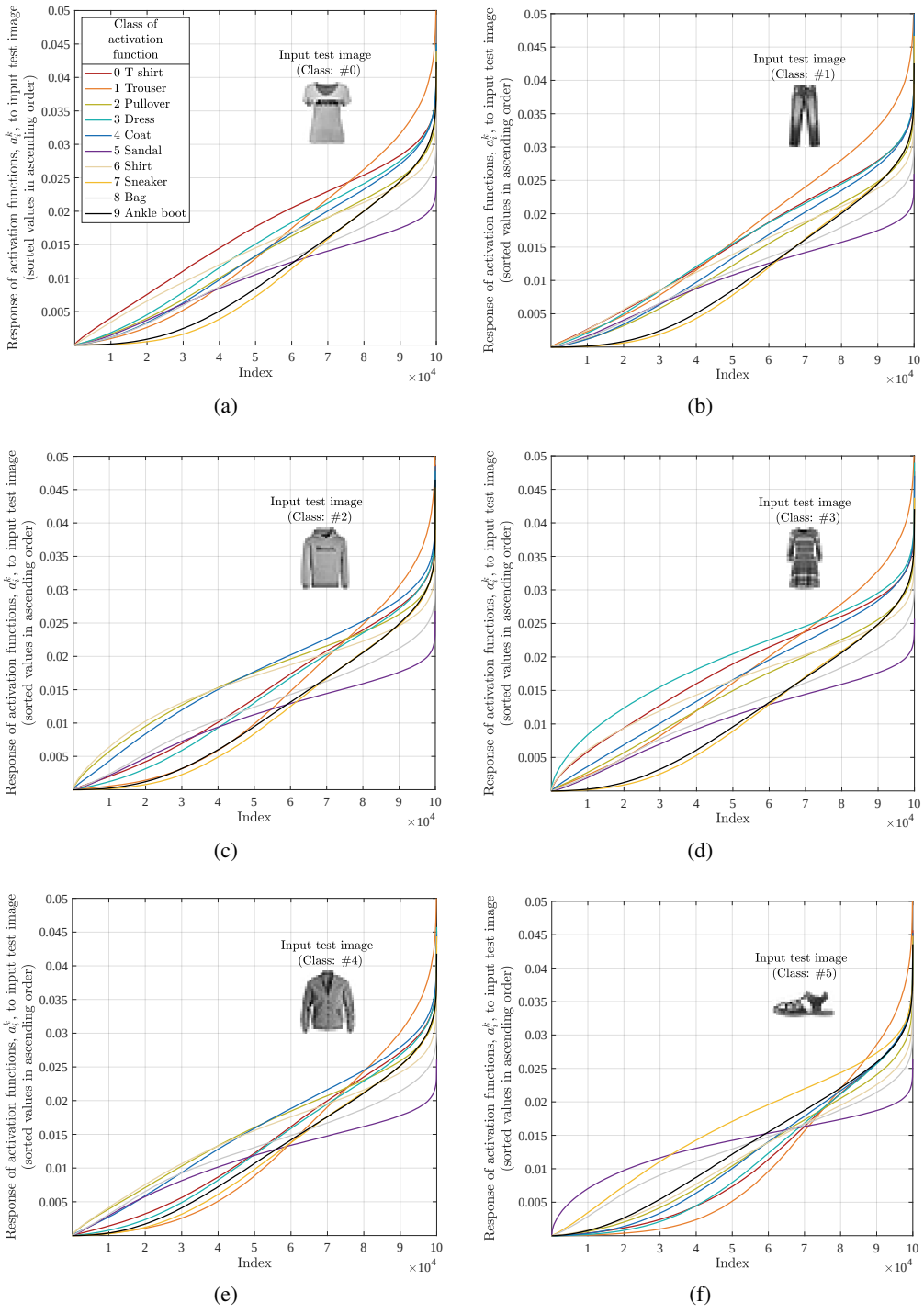
$$\mathop{\mathbb{E}}_{X\sim P(x|C_1)}\Big[P(x|C_1)P(C_1)\Big] \;>\; \mathop{\mathbb{E}}_{X\sim P(x|C_1)}\Big[P(x|C_0)P(C_0)\Big]$$

$$\frac{(\mu_1-\mu_0)^2}{\sigma_1^2+\sigma_0^2} \;>\; 2\log\left(\frac{\sqrt{2}P(C_0)}{P(C_1)}\right) - \log\left(1+\frac{\sigma_0^2}{\sigma_1^2}\right). \tag{19}$$

Combining inequalities 18 and 19, gives

$$\frac{(\mu_1-\mu_0)^2}{\sigma_1^2+\sigma_0^2} > 2\max\left\{\log\left(\frac{\sqrt{2}P(C_0)}{P(C_1)}\right) - \frac{1}{2}\log\left(1+\frac{\sigma_0^2}{\sigma_1^2}\right)\;\;,\;\; \log\left(\frac{\sqrt{2}P(C_1)}{P(C_0)}\right) - \frac{1}{2}\log\left(1+\frac{\sigma_1^2}{\sigma_0^2}\right)\right\} \tag{20}$$

E RESPONSES OF ACTIVATION FUNCTIONS



(a) (b) (c) (d) (e) (f)

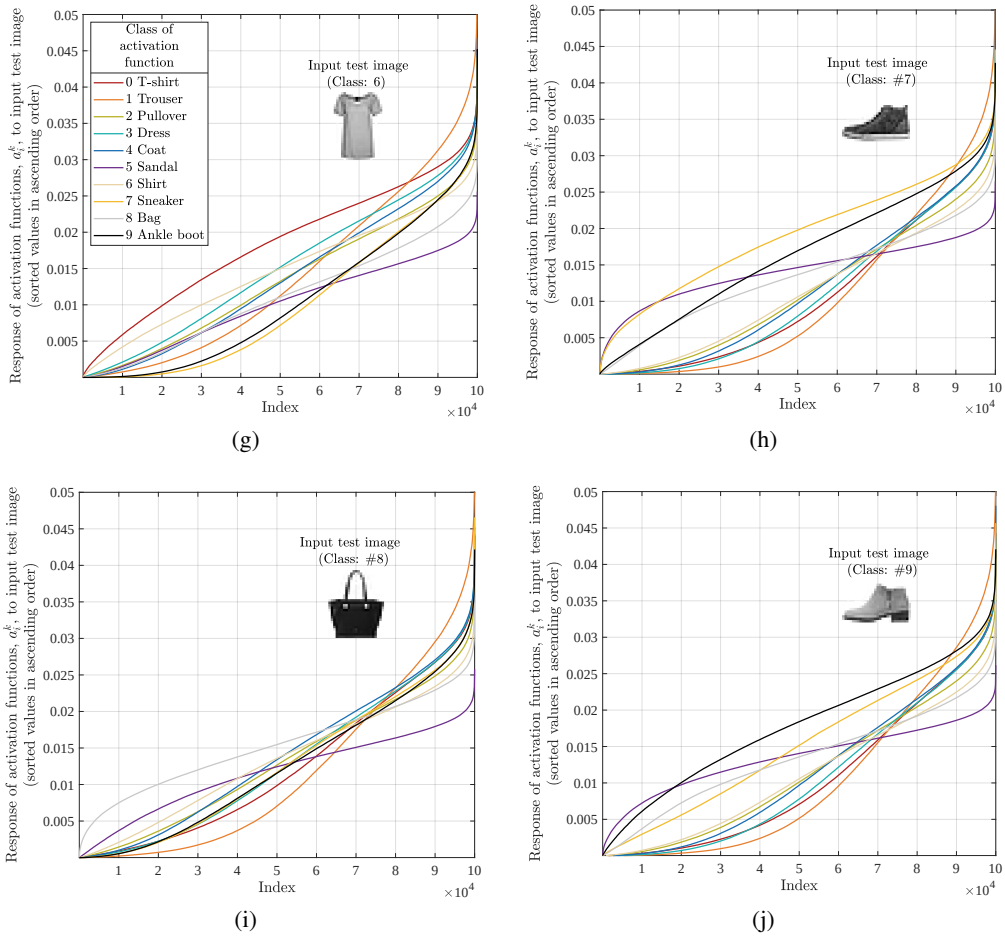


Figure 5: (a)–(j) Responses of activation functions, a_i^k, to unseen test images. The y-axis displays the sorted values of set $\{a_1^k, a_2^k, \ldots, a_m^k\}$ in ascending order, while x-axis is the index of said sorted values (see Fig. 1a for a schematic illustration depicting a_i^k responses). The number of nodes in the hidden layer of the network is $m = 10^5$.

F PERFORMANCE COMPARISONS

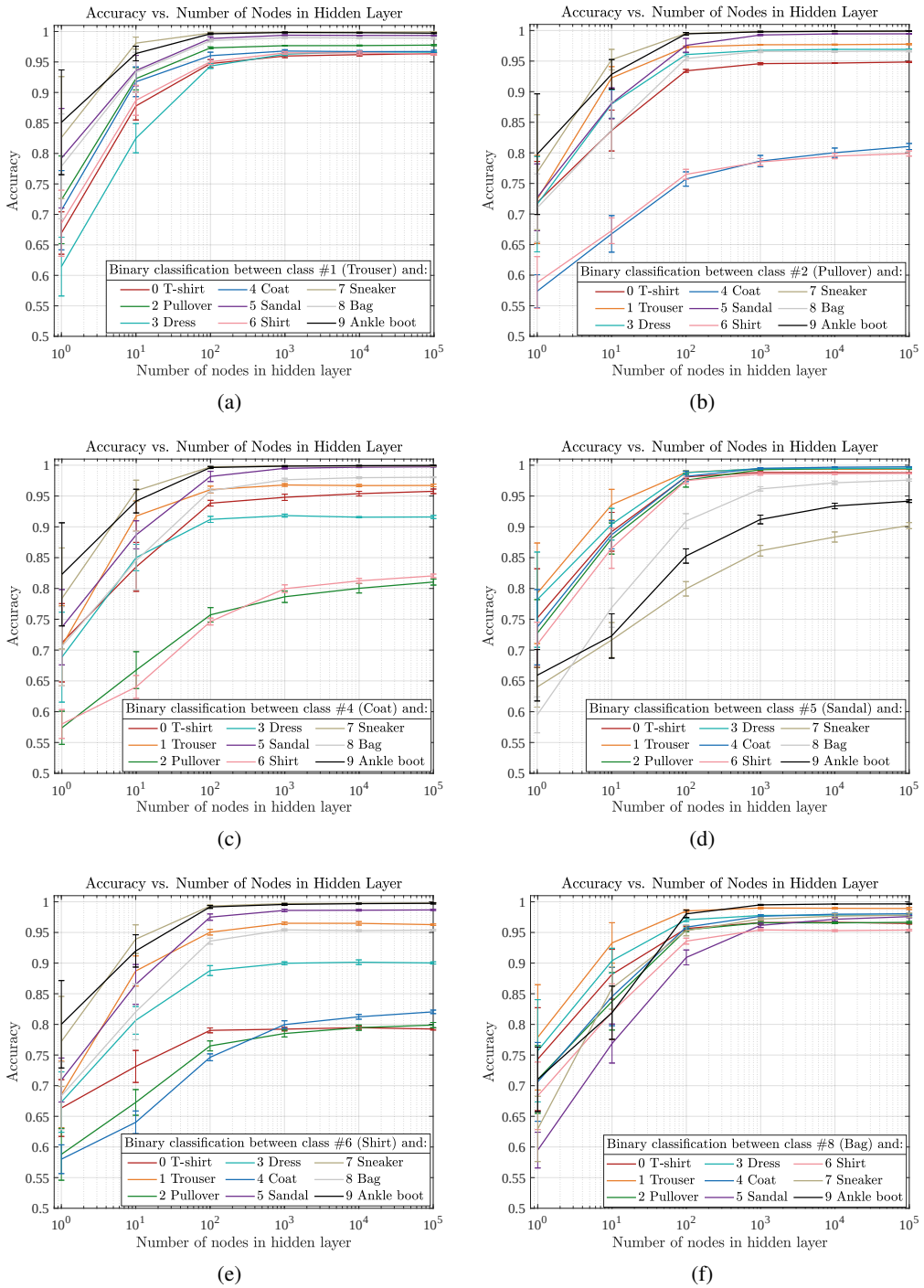


Figure 6: (a)–(f) Classification accuracy versus number of nodes, m, in the hidden layer of the neural network. Error bars are 95% confidence intervals and are computed by randomizing weights of the network a total of 10 times.

G ADDITIONAL ILLUSTRATIONS FOR PROPOSED SCHEME

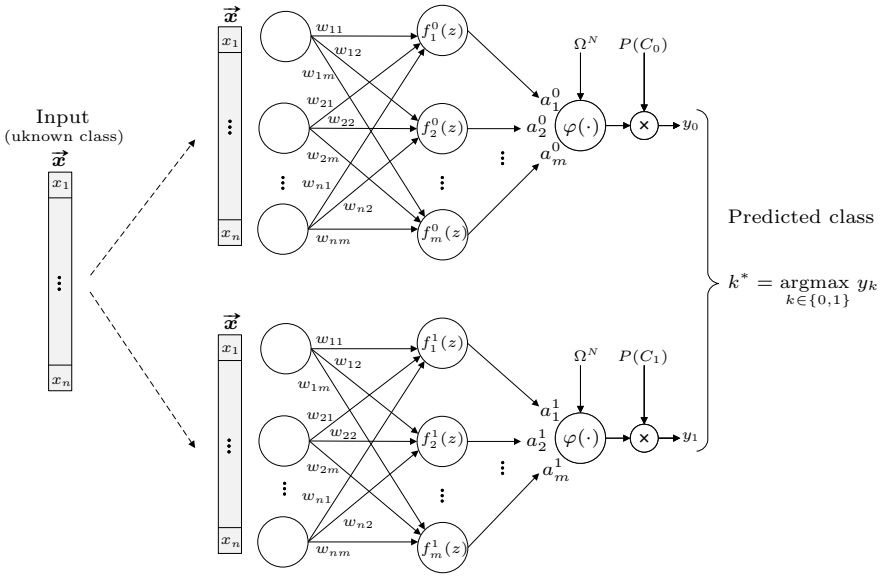


Figure 7: Expanded illustration of proposed scheme. The top network is for the first class ($k = 0$), while the bottom network is for the second class ($k = 1$). The weights of both networks are identical. An input sample of an unknown class is presented to both networks, and each network outputs a corresponding y_k value. The class of the network that produces the maximum value of y_k is declared the class of the input sample.

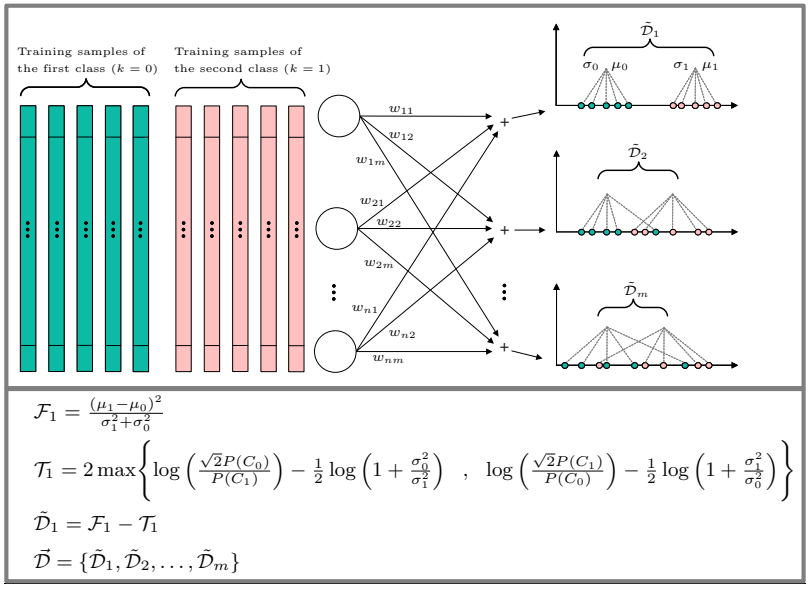


Figure 8: Illustration for the construction of set $\vec{\mathcal{D}}$. Consider the uppermost node in the network for which we need to compute $\tilde{\mathcal{D}}_1$. Here samples of each class are multiplied by random weights $w_1^T = [w_{11}, w_{21}, \ldots, w_{n1}]$. After which, the values of μ_k and σ_k are computed, $k = 0, 1$ (Section 5). Using μ_k and σ_k with $P(C_k)$, the value of the divergence at the node under consideration is $\tilde{\mathcal{D}}_1 = \mathcal{F}_1 - \mathcal{T}_1$. This process is repeated for all nodes in the network to obtain set $\vec{\mathcal{D}} = \{\tilde{\mathcal{D}}_1, \tilde{\mathcal{D}}_2, \ldots, \tilde{\mathcal{D}}_m\}$.