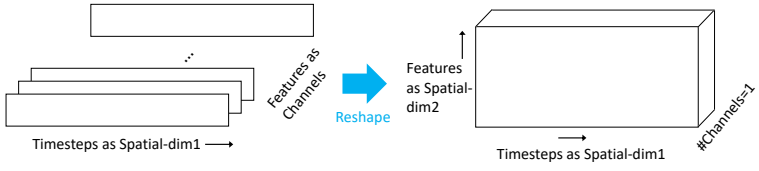


Figure 6.24. *A reshaped multivariate time series that resembles structure of an image with a single channel.*

a general problem in most deep learning networks.

6.10 Multivariate Time Series Modeled as Image

In the previous § 6.9, convolutional neural networks were constructed following the conventional approach of placing the temporal axis along a spatial axis and the multivariate features as channels.

Another approach is placing the features along a second spatial axis. This can be done by reshaping the input samples from `(timesteps, features)` to `(timesteps, features, 1)` tensors. The reshape is illustrated in Figure 6.24. The reshaped time series appears in the shape of an image with a single channel. Hence, this approach is termed as modeling a multivariate time series as an image.

In the reshaped time series sample, the `timesteps` and `features` are the spatial axes with one channel. Due to the two spatial axes, a convolutional network is constructed with `Conv2D`.

In the following, first, a `Conv2D` network equivalent to the baseline `Conv1D` network in § 6.9.3 is constructed to show their interchangeability in § 6.10.1. Thereafter, another network is constructed with `Conv2D` which is intended to learn the **local** temporal and spatial dependencies in and between the features in § 6.10.2.

6.10.1 `Conv1D` and `Conv2D` Equivalence

This section shows the equivalence between `Conv1D` and `Conv2D` by modeling the multivariate time series like an image.

In Listing 6.6, a convolutional network equivalent to the baseline network in § 6.9.3 is constructed using `Conv2D`. At its top, a lambda function to reshape the original samples is defined. It changes the `X` from `(samples, timesteps, features)` to `(samples, timesteps, features, 1)` tensor.

Listing 6.6. A `Conv2D` network equivalent to the baseline `Conv1D` network.

```
1  # Equivalence of conv2d and conv1d
2  def reshape4d(X):
3    return X.reshape((X.shape[0],
4                      X.shape[1],
5                      X.shape[2],
6                      1))
7
8  model = Sequential()
9  model.add(Input(shape=(TIMESTEPS,
10                        N_FEATURES,
11                        1),
12                  name='input'))
13 model.add(Conv2D(filters=16,
14                   kernel_size=(4, N_FEATURES),
15                   activation='relu',
16                   data_format='channels_last'))
17 model.add(MaxPool2D(pool_size=(4, 1)))
18 model.add(Flatten())
19 model.add(Dense(units=16,
20                  activation='relu'))
21 model.add(Dense(units=1,
22                  activation='sigmoid',
23                  name='output'))
24 model.summary()
```

Thereafter, a network with `Conv2D` is defined. The network is made equivalent to the `Conv1D` network by setting the `kernel_size` in line 14 as `(4, n_features)`. This kernel covers the entire feature axis which

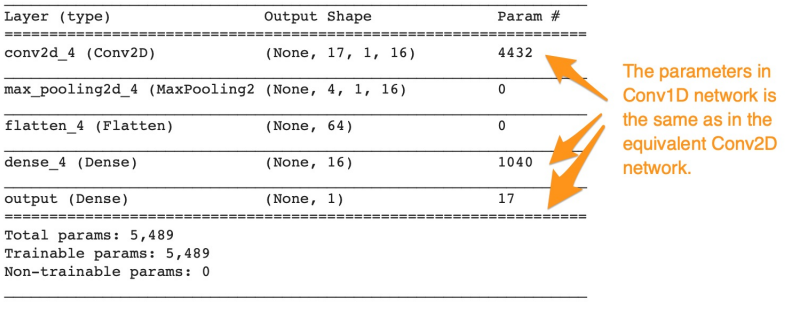


Figure 6.25. *Summary of a `Conv2D` equivalent to the baseline `Conv1D` network shows their interchangeability.*

makes the network the same as the `Conv1D` network.

They become the same because the features are the channels in the `Conv1D` network. Therefore, the `Conv` kernel spans all the channels. Setting the `Conv2D` kernel width equal to the number of features replicated this behavior. This is also confirmed by comparing the parameters in each layer shown in the model summary in Figure 6.25 with the baseline model summary in Figure 6.21. All of them are the same.

The result of the `Conv2D` network here is not shown as they resemble the baseline results in § 6.9.3.

The purpose of this section is to show the interchangeability of `Conv1D` and `Conv2D` in constructing convolutional networks. A practitioner can choose between either.

`Conv2D` networks provide more flexibility. Using it, one can construct models equivalent to a `Conv1D` network as well as try other architectures by treating a multivariate time series as an image. This is shown in the next section.

6.10.2 Neighborhood Model

A benefit of treating multivariate time series as an image for modeling is: a network can be constructed to learn the **local** temporal and spatial dependencies called a *neighborhood model*. These models require fewer convolutional parameters. One such network is developed in Listing 6.7.

Listing 6.7. Neighborhood model for multivariate time series

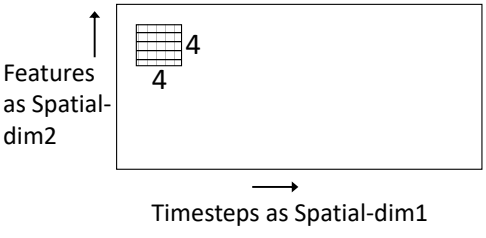
```
def reshape4d(X):
  return X.reshape((X.shape[0],
                    X.shape[1],
                    X.shape[2],
                    1))

# Neighborhood Model
model = Sequential()
model.add(Input(shape=(TIMESTEPS,
                       N_FEATURES,
                       1),
                name='input'))
model.add(Conv2D(filters=16,
                 kernel_size=(4, 4),
                 activation='relu',
                 data_format='channels_last',
                 name='Conv2d'))
model.add(MaxPool2D(pool_size=(4, 4),
                    name='MaxPool'))
model.add(Flatten(name='Flatten'))
model.add(Dense(units=16,
                activation='relu',
                name='Dense'))
model.add(Dense(units=1,
                activation='sigmoid',
                name='output'))
model.summary()
```

The `kernel_size` for the convolutional layer is set as `(4, 4)`. As visually illustrated in Figure 6.26a, the kernel can **only** learn the **local** dependencies within a 4×4 span and, hence, referred to as a *neighborhood model*.

Besides, as the reshaped time series has a single channel, the convolutional kernel size becomes `(4, 4, 1)` as opposed to `(4, n_features, 1)` in the previous § 6.10.1. Consequently, as shown in Figure 6.26b, the convolutional parameters reduce significantly.

However, due to the smaller kernel, the convolutional feature map becomes larger than in the baseline. This causes the penultimate dense layer parameters to increase.



(a) *A* 4×4 *kernel.* .

Layer (type)	Output Shape	Param #
Conv2d (Conv2D)	(None, 17, 66, 16)	272
MaxPool (MaxPooling2D)	(None, 4, 16, 16)	0
Flatten (Flatten)	(None, 1024)	0
Dense (Dense)	(None, 16)	16400
output (Dense)	(None, 1)	17

Total params: 16,689
Trainable params: 16,689
Non-trainable params: 0

Fewer Conv parameters due to smaller kernel in the neighborhood-model.

The Dense layer parameters are, however, significantly higher than the baseline model due to larger Conv feature map.

(b) *Neighborhood model summary.*

Figure 6.26. *In the top figure, the horizontal and vertical axes are time (i.e., the temporal dimension) and features (i.e., the spatial dimension), respectively. The* 4×4 *kernel spans both the axes and learns the local spatio-temporal dependencies. The constructed convolutional network summary (bottom) shows a significant reduction in the convolutional layer parameters compared to the baseline.*

A neighborhood model is expected to perform well if the interacting features are ordered or grouped, i.e., spatial dependencies are expected to be local. In such a scenario, it can have a similar performance as the baseline with fewer convolutional parameters that improve efficiency.

Besides, a neighborhood model also provides the flexibility to span a longer or the entire time-steps if long-term dependencies need to be learned. This is, otherwise, difficult if the time series is in its original shape as in the baseline model.

In sum, modeling a multivariate time series as an image brings more flexibility to construct efficient networks to learn short or wide spatial dependencies **and** short or long temporal dependencies based on the problem. The conventional modeling approach, on the other hand, only allows changing the length of the temporal dependencies.

6.11 Summary Statistics for Pooling

The strength of a convolutional network is its ability to simplify the feature extraction process. In this, pooling plays a critical role by weeding the extraneous information.

A pooling operation summarizes features into a *summary statistic*. It, therefore, relies on the statistic's efficiency. Whether the statistic preserves the relevant information or loses them depends on its efficiency.

What is an efficient summary statistic?

A summary statistic is a construct from *principles of data reduction* (Casella and Berger 2002). It is defined as,

> *a summary statistic summarizes a set of observations to preserve the* ***largest*** *amount of information as* ***succinctly*** *as possible.*

An efficient summary statistic is, therefore, one that concisely contains the most information of a sample. For example, the sample mean, or maximum. Other statistics, such as the sample skewness, or sample

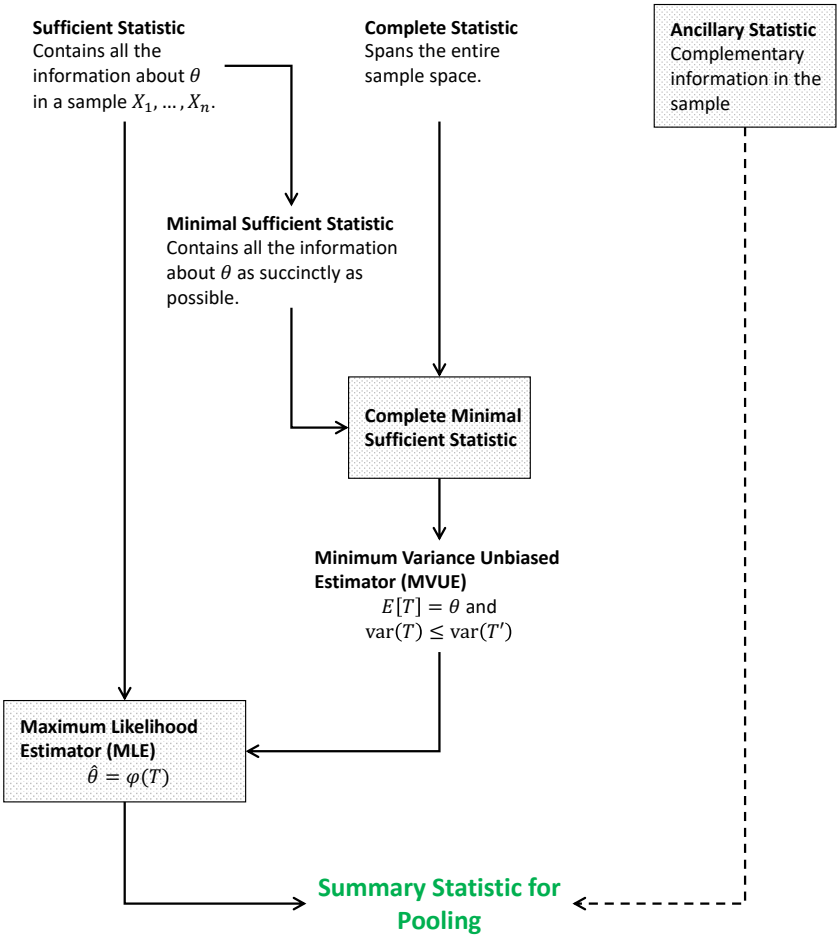


Figure 6.27. *A summary statistic for pooling has roots in sufficient, complete, and ancillary statistics. A statistic that is both sufficient and complete provides a minimum variance unbiased estimator (MVUE). An MVUE's properties make it an efficient statistic. For some distributions, the maximum likelihood estimator (MLE), e.g., sample maximum, is an MVUE and, hence, becomes the best pooling statistic. Moreover, ancillary statistics, such as sample range, extracts information complementary to the MLE. They can be used as an additional pooling statistic (see Appendix J).*

size do not contain as much relevant information and, therefore, not efficient for pooling.

This section lays out the theory of summary statistics to learn about efficient statistics for pooling.

> "An experimenter might wish to summarize the information in a sample by determining a few key features of the sample values. This is usually done by computing (summary) statistics—functions of the sample."
>
> –Casella and Berger 2002.

Learning the dependence of pooling on the efficiency of summary statistics and the theory behind them is rewarding. It provides answers to questions like,

- Currently, max-pool and average-pool are the most common. Could there be other equally or more effective pooling statistics?
- Max-pool is found to be robust and, hence, better than others in most problems. What is the cause of max-pool's robustness?
- Can more than one pooling statistic be used together? If yes, how to find the best combination of statistics?

This section goes deeper into the theory of extracting meaningful features in the pooling layer. In doing so, the above questions are answered. Moreover, the theory behind summary statistics also provides an understanding of appropriately choosing a single or a set of statistics for pooling.

Pooling operation computes a summary statistic and its efficacy relies on the efficiency of the statistic.

An efficient summary statistic is one that contains the most information in as few values as possible, e.g., the sample mean and variance.

In the following, summary statistics applicable to pooling from three categories: (minimal) sufficient statistics, complete statistics, and ancillary statistics are explained. First, a few definitions are given in § 6.11.1. Then, sufficient statistics are shown to contain all the sample information in § 6.11.2. Next, § 6.11.3 shows complete statistics span the entire sample and a complete sufficient statistic is the minimum variance unbiased statistic (MVUE). It is further shown that a distribution's maximum likelihood estimator (MLE), e.g., average and maximum, is complete, sufficient, and MVUE.

This means, MLEs span the entire sample, contains all the information as succinctly as possible and is efficient. Hence, they make the best pooling statistic. Moreover, ancillary statistics such as sample range are shown to have complementary information in § 6.11.4 which can improve a network if used as an additional pooling statistic.

The findings in this section are used later in § 6.12 and 6.13 to uncover discoveries such as the reason behind max-pool's superiority, the effect of nonlinear activation on pooling, and the MLEs of common distributions for pooling.

6.11.1 Definitions

The feature map outputted by a convolutional layer is the input to a pooling layer. The feature map is a random variable $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ where n is the feature map size[7].

An observation of the random variable is denoted as $\boldsymbol{x} = \{x_1, \ldots, x_n\}$. Describing properties of random variables is beyond the scope of this book but it suffices to know that their true underlying distribution and parameters are unknown[8].

The distribution function, i.e., the *pdf* or *pmf*[9], for the random variable $\boldsymbol{X}$ is denoted as f. The distribution has an underlying unknown parameter θ. The θ characterizes the observed $\boldsymbol{x}$ and, therefore, should

[7]The variables are denoted in block letters to denote they are random variables.

[8]Refer to Chapter 5 in Casella and Berger 2002 to learn the properties of random variables.

[9]Pdf or pmf refers to probability density function or probability mass function for continuous or discrete distributions, respectively.

be estimated.

A summary statistic of $f(\boldsymbol{X})$ is an estimate of θ. The statistic is a function of the random variable denoted as $T(\boldsymbol{X})$ and computed from the sample observations as $T(\boldsymbol{x})$. The sample mean, median, maximum, standard deviation, etc. are examples of the function T.

The goal is to determine T's that contain the most **information** of the feature map, achieve the most **data reduction**, and are the most **efficient**. These T's are the best choice for pooling in convolutional networks.

6.11.2 (Minimal) Sufficient Statistics

> "A *sufficient* statistic for a distribution parameter θ is a statistic that, in a certain sense, captures all the information about θ contained in the sample."
>
> –Casella and Berger 2002.

The concept of *sufficient* statistics lays down the foundation of data reduction by summary statistics. It is formally defined as follows.

Definition 1. ***Sufficient Statistic.*** *A statistic $T(\boldsymbol{X})$ is a sufficient statistic for θ if the sample conditional distribution $f(\boldsymbol{X}|T(\boldsymbol{X}))$ does not depend on θ.*

The definition can be interpreted as the conditional distribution of $\boldsymbol{X}$ given $T(\boldsymbol{X})$, i.e., $f(\boldsymbol{X}|T(\boldsymbol{X}))$, is independent of θ. This implies that in presence of the statistic $T(\boldsymbol{X})$ any remaining information in the underlying parameter θ is not required.

A sufficient statistic can replace the distribution parameter θ.

It is possible only if $T(\boldsymbol{X})$ contains all the information about θ available in $\boldsymbol{X}$. Therefore, $T(\boldsymbol{X})$ becomes a *sufficient* statistic to represent the sample in place of θ.

For example,

- **Mean**. The sample mean, $T(\boldsymbol{X}) = \bar{X} = \frac{\sum_i X_i}{n}$, is a sufficient statistic for a sample from a **normal** or **exponential** distribution.
- **Maximum**. The sample maximum, $T(\boldsymbol{X}) = X_{(n)}$, where $X_{(n)} = \max_i X_i, i = 1, \ldots, n$ is the n-th order statistic[10], is a sufficient statistic in a (truncated) **uniform** distribution or approximately in a **Weibull** distribution if its shape parameter is large.

The average-pool (`AvgPool`) and max-pool (`MaxPool`) indirectly originated from here. They are commonly used pooling methods. Between them, `MaxPool` is more popular. But, why? It is answered shortly in § 6.12.1. Before getting there, summary statistics are explored in the context of pooling.

Mean and maximum are sufficient statistics which indirectly led to the origin of `AvgPool` *and* `MaxPool`*.*

The *sufficiency* of a statistic is proved using the Factorization Theorem.

Theorem 1. ***Factorization Theorem****. A statistic $T(\boldsymbol{X})$ is sufficient if and only if functions $g(t|\theta)$ and $h(\boldsymbol{x})$ can be found such that $f(\boldsymbol{x}|\theta) = g(T(\boldsymbol{x})|\theta)h(\boldsymbol{x})$.*

The proofs for the sufficiency of the sample mean and maximum for normal and uniform distributions, respectively, are in Casella and Berger 2002 Chapter 6. However, it is worthwhile to look at sufficient statistics for a normal distribution to realize there are multiple sufficient statistics for a distribution.

Proposition 1. *If $X_1, \ldots, X_n$ are iid normal distributed $N(\mu, \sigma^2)$, the sample mean $\bar{x} = \frac{\sum_i^n x_i}{n}$ and sample variance $s^2 = \frac{\sum_i^n (x_i - \bar{x})^2}{(n-1)}$ are the sufficient statistics for μ and σ^2, respectively.*

Proof. The parameters for a normal distribution are $\theta = (\mu, \sigma^2)$. The joint pdf of the sample $\boldsymbol{X} = X_1, \ldots, X_n$ is,

[10] An order statistic denoted as $X_{(i)}$ is the i-th largest observation in a sample. Therefore, $X_{(n)}$ is the maximum of a sample.

$$f(\boldsymbol{x}|\mu,\sigma^2) = \prod_i^n (2\pi\sigma^2)^{-1/2} \exp\big(-(x_i-\mu)^2/(2\sigma^2)\big)$$
$$= (2\pi\sigma^2)^{-n/2} \exp\Big(-\sum_i^n (x_i-\mu)^2/(2\sigma^2)\Big) \qquad (6.12)$$

The pdf depends on the sample $\boldsymbol{x}$ through the two statistics $T_1(\boldsymbol{x}) = \bar{x}$ and $T_2(\boldsymbol{x}) = s^2$.

Thus, using the Factorization Theorem we can define $h(\boldsymbol{x}) = 1$ and

$$g(\boldsymbol{t}|\theta) = g(t_1, t_2|\mu,\sigma^2)$$
$$= (2\pi\sigma^2)^{-n/2} \exp\Big(-\big(n(t_1-\mu)^2 + (n-1)t_2\big)/(2\sigma^2)\Big)$$

We can now express the pdf as

$$f(\boldsymbol{x}|\mu,\sigma^2) = g(T_1(\boldsymbol{x}), T_2(\boldsymbol{x})|\mu,\sigma^2)h(\boldsymbol{x}).$$

Hence, by Factorization Theorem, $T(\boldsymbol{X}) = (T_1(\boldsymbol{X}), T_2(\boldsymbol{X})) = (\bar{X}, S^2)$ is a sufficient statistic for (μ,σ^2) in this normal model. □

Proposition 1 shows that a sample from normal distribution has more than one sufficient statistic, $\bar{x}$, and s^2. Similarly, a uniform distribution has the sample maximum $\max_i x_i$ and minimum $\min_i x_i$ as its sufficient statistics.

This tells that sufficient statistics are not unique in a distribution. There can be many. In fact, the entire ordered sample $T(\boldsymbol{X}) = \boldsymbol{X} = (X_{(1)}, \ldots, X_{(n)})$ is also a sufficient statistic.

Of course $T(\boldsymbol{X}) = \boldsymbol{X}$ is not much of a data reduction. But, out of the several sufficient statistics, which is better than the other?

The answer lies in the defined purpose of a summary statistic. The purpose is to achieve as much data reduction as possible without loss of information about the parameter θ.

Therefore, a sufficient statistic that achieves the most data reduction while retaining all the information about θ is preferable. Such a statistic is formally called a *minimal sufficient statistic.*

Definition 2. ***Minimal Sufficiency.*** *A sufficient statistic $T(\boldsymbol{X})$ is called a minimal sufficient statistic if for any other sufficient statistic $T'(\boldsymbol{X})$, $T(\boldsymbol{X})$ is a function of $T'(\boldsymbol{X})$, i.e., $T(\boldsymbol{X}) = f(T'(\boldsymbol{X}))$ for any $\boldsymbol{X} \in \mathcal{X}$.*

This can be interpreted as if any sufficient statistic $T'(\boldsymbol{X})$ can be reduced to $T(\boldsymbol{X})$, it means $T(\boldsymbol{X})$ provides more data reduction without losing information. For example, $T(\boldsymbol{X}) = \max_i X_i$ and $T'(\boldsymbol{X}) = X_1, \ldots, X_n$ are sufficient statistics where $T(\boldsymbol{X}) = \max_i X_i = \max T'(\boldsymbol{X})$.

Thus, $T(\boldsymbol{X})$ has the information of θ more succinctly than any other $T'(\boldsymbol{X})$. And, therefore, $T(\boldsymbol{X})$ becomes *minimally sufficient.*

Mathematically, minimal sufficiency can be proved using the following theorem.

Theorem 2. *Suppose there exists a statistic $T(\boldsymbol{x})$ such that for every two samples $\boldsymbol{x}$ and $\boldsymbol{y}$ the ratio of their pdfs $\frac{f(\boldsymbol{x}|\theta)}{f(\boldsymbol{y}|\theta)}$ is a constant independent of θ if and only if $T(\boldsymbol{x}) = T(\boldsymbol{y})$. Then $T(\boldsymbol{X})$ is a minimal sufficient statistic for θ.*

Using the theorem, the minimal sufficient statistics for $X \sim Normal$ and $X \sim Uniform$ are shown in the following propositions.

Proposition 2. *The sample mean $\bar{x} = \frac{\sum_i^n x_i}{n}$ and sample variance $s^2 = \frac{\sum_i^n (x_i - \bar{x})^2}{(n-1)}$ are the **minimal** sufficient statistics for μ and σ^2, respectively, if $X_1, \ldots, X_n$ are iid normal $N(\mu, \sigma^2)$.*

Proof. Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ are two samples, and their sample mean and variances are $(\bar{x}, s_{\boldsymbol{x}^2})$ and $(\bar{y}, s_{\boldsymbol{y}^2})$, respectively.

Using the pdf expression in Equation 6.12, the ratio of pdfs of $\boldsymbol{x}$ and $\boldsymbol{y}$ is,

$$\frac{f(\boldsymbol{x}|\mu,\sigma^2)}{f(\boldsymbol{y}|\mu,\sigma^2)} = \frac{(2\pi\sigma^2)^{-n/2}\exp\Big(-\big(n(\bar{x}-\mu)^2+(n-1)s_{\boldsymbol{x}^2}\big)/(2\sigma^2)\Big)}{(2\pi\sigma^2)^{-n/2}\exp\Big(-\big(n(\bar{y}-\mu)^2+(n-1)s_{\boldsymbol{y}^2}\big)/(2\sigma^2)\Big)}$$
$$= \exp\Big(\big(-n(\bar{x}^2-\bar{y}^2)+2n\mu(\bar{x}-\bar{y})-(n-1)(s_{\boldsymbol{x}^2}-s_{\boldsymbol{y}^2})\big)/(2\sigma^2)\Big).$$

The ratio is a constant, i.e., independent of μ and σ, if and only if $\bar{x}=\bar{y}$ and $s_{\boldsymbol{x}^2}=s_{\boldsymbol{y}^2}$. Thus, by Theorem 2, $(\bar{(X)}, S^2)$ is minimal sufficient statistic for (μ,σ^2). □

Proposition 3. *The sample maximum* $\max_i X_i$ *and minimum* $\min_i X_i$ *are the* ***minimal*** *sufficient statistics for* θ *if* $X_1,\ldots,X_n$ *are iid uniform in the interval* $(-\theta,\theta)$*, and* $-\infty<\theta<\infty$.

Proof. The joint pdf of $\boldsymbol{X}$ from $U(-\theta,\theta)$ is,

$$f(\boldsymbol{x}|\theta) = \prod_i^n \frac{1}{2\theta}\mathbb{1}(|x_i|<\theta)$$
$$= \frac{1}{(2\theta)^n}\mathbb{1}(\max_i x_i<\theta)\cdot\mathbb{1}(\min_i x_i>-\theta)$$

Therefore, a ratio of pdfs of two samples $\boldsymbol{x}$ and $\boldsymbol{y}$ is,

$$\frac{f(\boldsymbol{x}|\theta)}{f(\boldsymbol{y}|\theta)} = \frac{\mathbb{1}(\max_i x_i<\theta)\cdot\mathbb{1}(\min_i x_i>-\theta)}{\mathbb{1}(\max_i y_i<\theta)\cdot\mathbb{1}(\min_i y_i>-\theta)}$$

The ratio is a constant independent of θ if and only if $\max_i x_i = \max_i y_i$ and $\min_i x_i = \min_i y_i$.

Therefore, by Theorem 2, $T(\boldsymbol{X}) = (\max_i X_i, \min_i X_i)$ is a minimal sufficient statistic for θ.

□

In sum, sufficient statistics provide all information about a sample. However, there are many sufficient statistics and most of them do not result in data reduction. A minimal sufficient statistic, on the other hand, preserves the information and provides as much data reduction as possible. Therefore, among the several choices of sufficient statistics, *minimal sufficient statistic(s)* should be taken for pooling.

*A minimal sufficient statistic such as mean and maximum has **all** the information about underlying distribution parameter θ present in a feature map as succinctly as possible.*

Moreover, any one-to-one mapping of a minimal sufficient statistic is also a minimal sufficient statistic. This is important knowledge. Based on this, a pooling statistic can be scaled to stabilize a network without affecting the statistic's performance. For example, one should be pooling with $\sum_i X_i/n$ and $\sqrt{\sum_i (X_i - \bar{X})^2/n}$ instead of $\sum_i X_i$ and $\sum_i (X_i - \bar{X})^2/n$, respectively.

Identifying the best one-to-one mapping is, however, not always straightforward. The approach to finding the best-mapped statistic is formalized by connecting minimal sufficient statistics with the maximum likelihood estimator (MLE) through the theory of complete statistics in the next section.

6.11.3 Complete Statistics

The many choices with minimal sufficient statistics sometimes confuse a selection. This section introduces complete statistics which narrows the pooling statistic choice to only the **maximum likelihood estimator** of the feature map distribution.

A complete statistic is a bridge between minimal sufficient statistics and MLEs. MLEs derived from complete minimal statistics have the essential attributes of unbiasedness and minimum variance along with the minimality and completeness properties. MLEs, therefore, become the natural choice for pooling. Thereby, removing most of the ambiguity

around pooling statistic selection.

In the following, these attributes and the path that leads to the relationship between complete minimal statistics and the MLE is laid out.

Definition 3. ***Completeness.*** *Let $f(t|\theta)$ be a family of pdfs or pmfs for a statistic $T(\boldsymbol{X})$. The family of probability distributions is called complete if for every measurable, real-valued function g, $E_\theta(g(T)) = 0$ for all $\theta \in \Omega$ implies $g(T) = 0$ with respect to θ, i.e., $P_\theta(g(T) = 0) = 1$ for all θ. The statistic T is boundedly complete if g is bounded.*

In simple words, it means a probability distribution is complete if the probability of a statistic $T(\boldsymbol{X})$ from an observed sample $\boldsymbol{X} = X_1, \ldots, X_n$ in the distribution is always non-zero.

This is clearer by considering a discrete case. In this case, completeness means $E_\theta(g(T)) = \sum g(T)P_\theta(T = t) = 0$ implies $g(T) = 0$ because by definition $P_\theta(T = t)$ is non-zero.

For example, suppose $X_1, \ldots, X_n$ is observed from a normal distribution $N(\mu, 1)$, and there is a statistic $T(\boldsymbol{X}) = \sum X_i$. Then, the $P_\mu(T(\boldsymbol{X}) = 0)$ is not equal to 0 for all μ. Therefore, $E_\mu(g(T)) = \int g(T)P_\mu(T) = 0$ implies $g(T) = 0$ for all μ. Therefore, $T = \sum X_i$ is complete.

This is an important property because it confirms that a statistic T, if complete, will span the whole sample space. Simply put, the statistic will contain *some* information from every observed sample X_i of the distribution for any parameter θ. And, therefore, the statistic is called **complete**.

A complete statistic contains some information about every observation from a distribution.

The importance of the *completeness* property is understood better by differentiating it with a sufficient statistic.

A minimal sufficient statistic contains **all** the information about θ, it does not necessarily **span** the whole sample space.

For example, suppose $X_1, \ldots, X_n$ is iid $Uniform(-\theta, \theta)$ then $T(\boldsymbol{X}) =$

$(X_{(1)}, X_{(n)})$, where $X_{(1)} = \min_i X_i$ and $X_{(n)} = \max_i X_i$ is a sufficient statistic. But it is **not** complete because $E(X_{(n)} - X_{(1)}) = c$, where c is a constant independent of θ. Therefore, we can define $g(T) = X_{(n)} - X_{(1)} - c$ but $E(X_{(n)} - X_{(1)} - c) = 0$ does not necessarily imply $X_{(n)} - X_{(1)} - c$ is always 0 because $E(X_{(n)} - X_{(1)}) \neq c$ for $\theta' \neq \theta$.

However, for the $Uniform$ distribution, $T = X_{(n)}$ is sufficient and complete. The proof is in § 6.2 in Casella and Berger 2002. It means $T = X_{(n)}$ spans the whole sample space.

For a normal distribution $N(\mu, \sigma^2)$, $T = (\sum_i X_i, \sum_i X_i^2)$ is both sufficient and complete. Meaning, the T has all the information about μ, σ^2 in a sample $X_1, \ldots, X_n$ as well as spans the whole sample space.

On a side note, a complete *statistic* is a misleading term. Instead of a statistic, completeness is a property of its family of distribution $f(t|\theta)$ (see Casella and Berger 2002 p.285). That means, when a statistic's distribution is complete it is called a *complete statistic*.

Next, the following Theorem 3 and 4 establish a relation between a complete statistic and a minimal sufficient statistic.

Theorem 3. ***Bahadur's theorem***[11]. *If T is a boundedly complete sufficient statistic and finite-dimensional, then it is minimal sufficient.*

A boundedly complete statistic in Theorem 3 implies the arbitrary function g in Definition 3 is bounded. This is a weak condition which is almost always true. Therefore, a complete sufficient statistic in most cases are also minimal.

The reverse, however, is always true as stated in Theorem 4.

Theorem 4. ***Complete Minimal Sufficient Statistic***[12]. *If a minimal sufficient statistic exists, then any complete sufficient statistic is also minimal.*

A complete minimal sufficient statistic has both **completeness** and **minimality** attributes. The statistic, therefore, **spans** the entire sample space, draws information from there, and yields **all** the information

[11] See Bahadur 1957.

[12] See § 6.2 in Casella and Berger 2002 and § 2.1 in Schervish 2012.

about the feature map distribution parameter θ as **succinctly** as possible. These attributes might appear enough, but are they?

They are not. Consider a complete minimal sufficient statistic $U = \sum_i X_i$ for a normally distributed feature map, $N(\mu, \sigma^2)$. Its expected value is $E(U) = n\mu$, which makes it biased. Using such a statistic in pooling can also make a convolutional network biased.

The biasedness in T is removed in $U = \frac{\sum_i X_i}{n}$. But there are other unbiased statistics as well, e.g., $U' = (X_{(1)} + X_{(n)})/2$. Which among them is better for pooling? The one with a smaller variance.

Compare the variances of U and U': $var(U) = \sigma^2/n$ and $var(U') = \sigma^2/2$. Clearly, $var(U) < var(U')$, if $n > 2$. It means if suppose U' is used in pooling, its value will swing significantly from sample to sample. This makes the network training and inferencing unstable. U, on the other hand, will have smaller variations that bring model stability.

Unbiasedness and small variation, therefore, in a pooling statistic makes a convolutional network efficient. This brings us to another type of statistic called minimum variance unbiased estimator (MVUE). It is defined as,

Definition 4. ***Minimum Variance Unbiased Estimator (MVUE).*** *A statistic T is a minimum variance unbiased estimator if T is unbiased, i.e., $E(T) = \theta$ and $var(T) \leq var(T')$ for all unbiased estimator T' and for all θ. Due to unbiasedness and small variance, it is statistically efficient.*

An MVUE is of particular interest due to its efficiency. Using an MVUE instead of any other statistic is analogous to using scaled input in a deep learning network. Just like an unscaled input, a biased and/or high variance pooling statistic makes the network unstable.

Identification of an MVUE provides an efficient statistic. Also, to our benefit, MVUE is unique. This is vital because the lookout for the best pooling statistic is over once the MVUE is found for a feature map. The uniqueness property is given in Theorem 5 below.

Theorem 5. ***MVUE is unique****. If a statistic T is a minimum variance unbiased estimator of θ then $T$is unique.*

Proof. Suppose T' is another MVUE $\neq T$. Since both T and T' are MVUE, their expectations will be θ, i.e., $E(T) = E(T') = \theta$, and the lowest variance denoted as δ, i.e., $var(T) = var(T') = \delta$.

We define an unbiased estimator combining T and T' as,

$$T^* = \frac{1}{2}(T + T').$$

For T^*, we have

$$E(T^*) = \theta$$

and,

$$\begin{aligned} var(T^*) &= var\left(\frac{1}{2}T + \frac{1}{2}T'\right) \\ &= \frac{1}{4}var(T) + \frac{1}{4}var(T') + \frac{1}{2}cov(T, T') \\ &\leq \frac{1}{4}var(T) + \frac{1}{4}var(T') + \\ &\frac{1}{2}(var(T)var(T'))^{1/2} && \text{Cauchy-Schwarz inequality} \\ &= \delta && \text{As, } var(T) = var(T') = \delta. \end{aligned}$$

But if the above inequality is strict, i.e., $var(T^*) < \delta$, then the minimality of δ is contradicted. So we must have equality for all θ.

Since the inequality is from Cauchy-Schwarz, we can have equality iff,

$$T' = a(\theta)T + b(\theta).$$

Therefore, the covariance between T and T' is,

$$\begin{aligned} cov(T, T') &= cov(T, a(\theta)T + b(\theta)) \\ &= cov(T, a(\theta)T) \\ &= a(\theta)var(T) \end{aligned}$$

but from the above equality $cov(T, T') = var(T)$, therefore, $a(\theta) = 1$. And, since

$$\begin{aligned} E(T') &= a(\theta)E(T) + b(\theta) \\ &= \theta + b(\theta) \end{aligned}$$

should be θ, $b(\theta) = 0$.

Hence, $T' = T$. Thus, T is unique.

□

Due to the uniqueness and efficiency properties, the MVUE statistic is an ideal statistic in pooling and, therefore, should be identified. As shown in Figure 6.27, a complete minimal sufficient statistic leads to the MVUE as per Theorem 6 below.

Theorem 6. ***Lehmann-Scheffé***[13]. *Let T be a complete (minimal) sufficient statistic and there is any unbiased estimator U of θ. Then there exists a unique MVUE, which can be obtained by conditioning U on T as $T^* = E(U|T)$. The MVUE can also be characterized as a unique unbiased function $T^* = \varphi(T)$ of the complete sufficient statistic.*

A complete (minimal) sufficient statistic does not guarantee unbiasedness and low variance by itself. However, Theorem 6 tells that a one-to-one function of it is an MVUE and is, of course, complete and minimal.

Therefore, an MVUE statistic T^*, if found, has both efficiency and complete minimal statistic properties. These properties make it supreme for pooling. However, a question remains, how to find the T^*?

Theorem 7 leads us to the answer in Corollary 1.

Theorem 7. ***MLE is a function of sufficient statistic.*** *If T is a sufficient statistic for θ, then the maximum likelihood estimator (MLE) (if it exists) $\hat{\theta}$ is a function of T, i.e., $\hat{\theta} = \varphi(T)$.*

[13]See Lehmann and Scheffé 1950 and Lehmann and Scheffé 1955.

Proof. Based on the Factorization Theorem 1,

$$f(\boldsymbol{x}|\theta) = g(T(\boldsymbol{x}|\theta))h(\boldsymbol{x}).$$

An MLE is computed by finding the θ that maximizes the likelihood function $L(\theta|\boldsymbol{x}) = f(\boldsymbol{x}|\theta)$.

If MLE is unique then $h(\boldsymbol{x})$ is a constant or equal to 1 without loss of generality. Therefore,

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} f(\boldsymbol{x}|\theta) \\ &= \arg\max_{\theta} g(T(\boldsymbol{x}|\theta))\end{aligned}$$

which is clearly a function of the sufficient statistic $T(\boldsymbol{X})$. □

Corollary 1. *If MLE $\hat{\theta}$ is unbiased and a complete minimal sufficient statistic T exist for parameter θ, then $\hat{\theta} = \varphi(T)$, and it is the unique minimum variance unbiased estimator (MVUE). Thereby, the statistic $\hat{\theta}$ contains all the information about the parameter θ, spans the whole sample space, and is efficient.*

Proof. Based on Theorem 7, the maximum likelihood estimator (MLE) $\hat{\theta}$ is a function of a sufficient statistic T, i.e., $\hat{\theta} = \varphi(T)$.

If T is complete sufficient and $\hat{\theta}$ is unbiased, then based on Theorem 6, $\hat{\theta} = \varphi(T)$ and is an MVUE. Therefore, being a function of complete statistic, $\hat{\theta}$ spans the whole sample space. Also, it is efficient based on the definition of MVUE in Definition 4.

Based on Theorem 3 and Theorem 4 a complete statistic in most cases is minimal, and if a minimal statistic exist then any complete statistic is always minimal. Therefore, if T is complete minimal, $\varphi(T)$ is also complete minimal and, therefore, $\hat{\theta} = \varphi(T)$ will have all the information about θ as succinctly as possible.

Finally, as per Theorem 5 an MVUE is unique. Therefore, $\hat{\theta}$ will be unique. □

At this stage, Theorem 3-7 come together to forge a spectacular relationship in Corollary 1. This is a pivotal result because a maximum likelihood estimator is available for most of the distributions applicable to the convolutional feature maps inputted to pooling.

MLE's properties elucidated in Corollary 1 make it an obvious choice in pooling. It is shown in § 6.13 that *average* and *maximum* pooling statistics are indeed MLEs. In fact, a parameterization combining the average and maximum in Boureau, Ponce, and LeCun 2010 to obtain a pooling statistic between the two is shown to be the MLE for Weibull distribution.

6.11.4 Ancillary Statistics

A (complete) minimal sufficient statistic itself or in the form of MLE retains all the information about the parameter θ. It eliminates all the extraneous information in the sample and takes only the piece of information related with θ.

Therefore, it might be suspected that no more information remains to draw from the sample. Except that there is more.

There is still information remaining in the sample that is independent of θ. For example, sample range or interquartile range. Such statistics are called *ancillary statistics*.

Ancillary statistics, defined below, have information complementary to minimal sufficient statistics. They can, therefore, act as a strong companion to a minimal sufficient based MLE statistic in pooling.

Definition 5. ***Ancillary Statistic.*** *A statistic $S(\boldsymbol{X})$ whose distribution does not depend on the parameter θ is called an ancillary statistic.*

As mentioned above, the sample range or interquartile range are examples of ancillary statistics. Both are special cases of a range statistic $R = X_{(k)} - X_{(l)}$, where $X_{(k)}$ and $X_{(l)}$ are order statistics with $k, l \in \{1, \ldots, n\}$. Proposition 4 shows that a range statistic is an ancillary statistic for any location model, e.g., normal and uniform distribution.

Proposition 4. *If $X_1, \ldots, X_n$ are iid observations from a location model where cdf is denoted as $F(x - \theta), -\infty < \theta < \infty$, e.g., Uniform and*

Normal, then any range statistic $R = X_{(k)} - X_{(l)}$ *is an ancillary statistic, where* $X_{(k)}$ *and* $X_{(l)}$ *are order statistics with* $k, l \in \{1, \ldots, n\}$*. The sample range* $R = X_{(n)} - X_{(1)} = \max_i X_i - \min_i X_i$ *and inter-quartile range* $R = Q_3 - Q_1$ *are special cases of a range.*

Proof. We have $X \sim F(X - \theta)$. We replace X with Z such that $X = Z + \theta$. Thus, the cdf of a range statistic $R = X_{(k)} - X_{(l)}$ becomes,

$$\begin{aligned} F_R(r|\theta) &= P_\theta(R \leq r) \\ &= P_\theta(X_{(k)} - X_{(l)} \leq r) \\ &= P_\theta((Z_{(k)} + \theta) - (Z_{(l)} + \theta) \leq r) \\ &= P_\theta(Z_{(k)} - Z_{(l)} + \theta - \theta \leq r) \\ &= P_\theta(Z_{(k)} - Z_{(l)} \leq r) \end{aligned}$$

The cdf of R, therefore, does not depend on θ. Hence, the range statistic R is an ancillary statistic. □

As per the proposition, a range statistic can be used in conjunction with any other minimal sufficient statistic in pooling. However, the combination should be chosen carefully. They are sometimes dependent. For example, in a Uniform model, the minimal sufficient statistics are $T_1(\boldsymbol{X}) = \max_i X_i$ and $T_2(\boldsymbol{X}) = \min_i X_i$, and an ancillary statistic is $S(\boldsymbol{X}) = \max_i X_i - \min_i X_i$. Clearly, $S(\boldsymbol{X})$ is a function, and, thereof dependent, of $T_1(\boldsymbol{X})$ and $T_2(\boldsymbol{X})$.

A complete minimal statistic, however, is independent of any ancillary statistic as per Theorem 8.

Theorem 8. ***Basu's Theorem***[14]. *If* $T(\boldsymbol{X})$ *is complete and minimal sufficient statistic, then* $T(\boldsymbol{X})$ *is independent of every ancillary statistic* $S(\boldsymbol{X})$*.*

Therefore, a minimum variance unbiased MLE based off of a complete minimal statistic is always independent of an ancillary statistic.

[14]See Basu 1955.

This property reinforces the support for using MLE as the primary pooling statistic. And, if needed, an ancillary statistic can be directly included due to their mutual independence. For illustration, Appendix J develops a Convolutional Network with *maximum* (MLE) pool and *sample range* (ancillary statistic) pool put together in parallel.

The ancillary statistic in pooling can draw additional relevant information from the convolutional feature map to improve a network's performance.

6.12 Pooling Discoveries

A convolutional network mainly comprises of three operations of convolution, activation, and pooling. Among them, pooling plays a key role in *extracting the essence from the excess* to improve the computational and statistical efficiencies.

There are a variety of pooling statistics developed over the years and discussed in § 6.15. Despite the variety, max-pooling remains popular due to its superior performance in most data sets.

The upcoming § 6.12.1 puts forward a plausible reason behind max-pool's superiority. The expounded reasoning also uncovers an inherent fallacy of distribution distortion in the *convolution* → *activation* → *pooling* structure in traditional networks.

Remedial architectures from Ranjan 2020 to address the fallacy by preserving the features map distribution is in § 6.12.2. The distribution preservation leads to presenting maximum likelihood estimators (MLEs) for pooling in § 6.13 (based on Corollary 1).

The section also shows a unifying theory behind max- and average-pooling, and their combination as mixed pooling in the form of MLE statistic of a Weibull distribution. Thereafter, a few advanced pooling techniques based off of summary statistics to adopt adaptive pooling and address spatial relationships are laid in § 6.14.

Lastly, the history of pooling discussed in § 6.15 ties together the

literature with the rest of this section.

6.12.1 Reason behind Max-Pool Superiority

The realization of max-pooling importance traces back to biological research in Riesenhuber and Poggio 1998. Riesenhuber and Poggio 1999 provided a biological explanation of max-pool superiority over average. Popular work by deep learning researchers have also advocated for max-pooling in Yang et al. 2009; Boureau, Ponce, and LeCun 2010; Saeedan et al. 2018.

Yang et al. 2009 reported significantly better classification performance on several object classification benchmarks using max-pooling compared to others.

A theoretical justification was provided in Boureau, Ponce, and LeCun 2010. They provided a theory supporting max-pool assuming the input to pooling as Bernoulli random variables. But the Bernoulli assumption is an extreme simplification. A traditional features map input to pooling is a continuous random variable while Bernoulli is discrete.

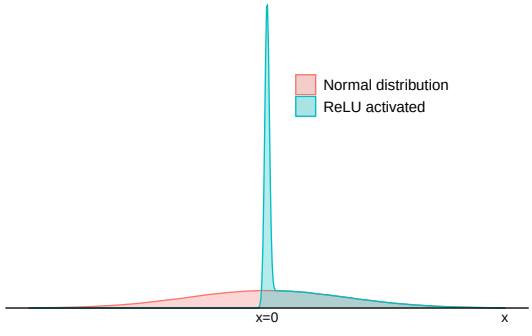
The reason for max-pool's superiority lies in the understanding distribution of feature maps, or rather the distorted distribution and its MLE statistic.

A feature map is a continuous random variable. A random variable follows a distribution. The MLE of the distribution, if known, is the best pooling statistic. The question is, what is the distribution of the feature map?

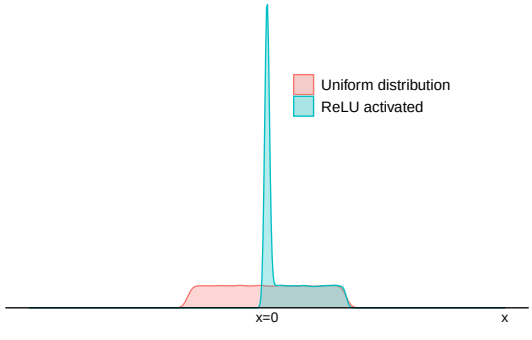
It depends on the problem. However, determining the distribution is not difficult. A bigger issue is that a feature map's distribution is already distorted before pooling. This is caused by a nonlinear activation of convolution output.

A nonlinear activation **distorts** the original distribution. Fitting known distributions on them become difficult. This is illustrated in Figure 6.28a. The figure shows the effect of ReLU activation on a normally distributed feature map.

As shown in the figure, the original distribution warps due to ReLU activation. It becomes severely skewed and does not follow a known



(a) *Normal Distribution before and after ReLU activation.*



(b) *Uniform Distribution before and after ReLU activation.*

Figure 6.28. *The feature map outputted from a convolutional network typically follows a distribution. The true distribution is, however, distorted by a nonlinear activation. For example, ReLU activated normal and uniform distribution shown here are severely skewed. Due to this, the feature map becomes biased. Therefore, most summary statistics other than the maximum becomes unusable for pooling. For example, the sample average yields an overestimate of the mean, the minimum statistic remains zero irrespective of the true smallest value, and a range statistic becomes =(maximum − zero). In effect, a nonlinear activation before pooling restricts its ambit, i.e., only a few summary statistics in pooling remain usable.*

distribution. If it is still assumed as a normal distribution, the sample mean (the MLE) will be an overestimate of the true mean. The overestimation undermines the average statistic for pooling. Similarly, other statistics, if used, in pooling such as the sample variance (an ancillary statistic[15]) will be underestimated.

The average statistic is overestimated under the normality assumption due to the distortion. This explains the reason behind average-pooling unfitness in some problems.

How does the maximum pooling remain effective in presence of the distortion? There are two plausible reasons based on distribution assumptions described below.

- **Uniform distribution**. The maximum statistic is the MLE of a uniform distribution. As shown in Figure 6.28b, the distortion does not affect the sample maximum. Although an activated feature map is no longer uniform if it was originally uniformly distributed the maximum statistic remains undisturbed for pooling.

- **Weibull distribution**. An activated feature map can be fitted with a Weibull distribution. It is quite a flexible distribution. It has various shapes for different values of its scale λ and shape k parameters. A few illustrations for different (λ, k) are shown in Figure 6.34 in § 6.13.4. The section also presents Weibull's MLE in Equation 6.25 which becomes equal to the sample maximum for large k.

Under these conditions, the sample maximum becomes the best pooling statistic. Perhaps, most of the problems are close to one of them and, hence, max-pooling is popular.

Max-pool is robust to the distortions in feature map distribution caused by nonlinear activation. Therefore, it works better than other types of pooling in most problems.

[15]Sample variance is an ancillary statistic as well as the MLE of normal distribution's variance parameter

The above reasons are conjectures. The exact theoretical reasoning behind max-pool's superiority is still elusive. And, as the theory behind pooling evolves, a better pooling statistic backed with its theoretical efficiency might be discovered.

6.12.2 Preserve Convolution Distribution

The premise of max-pool's superiority also uncovered a traditional convolutional network fault: convolution feature map distribution distortion before pooling. Distortion of feature map distribution is detrimental to pooling because the information is lost or masked.

This fault is present due to the structure *convolution* → *activation* → *pooling* in traditional networks. The structure follows the deep learning paradigm of nonlinear activation following a trainable layer.

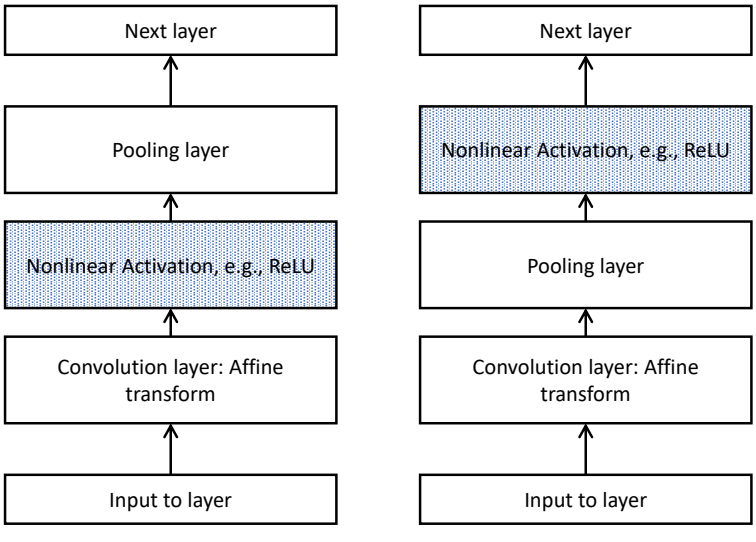
Nonlinear activation on convolutional layer output before pooling is an architectural fault.

But following this paradigm is not necessary for convolutional networks. A pooling layer is not trainable[16]. Therefore, nonlinear activation before or after pooling does not affect the overall nonlinearity of a convolutional network. Thus, their positions can be swapped to follow a structure *convolution* → *pooling* → *activation* to preserve the feature map's distribution while not impacting the network's nonlinearity.

Ranjan 2020 made the above argument and proposed the swapping. Swapping the order of pooling and nonlinear activation remedies the distribution distortion fallacy. The difference between the traditional and a swapped architecture is shown in Figure 6.29a and 6.29b.

Due to the swapping, the information derived in the convolution operation remains intact for pooling. Importantly, the convolutional feature map's distribution remains undisturbed. This brings a few major improvements,

[16] A pooling layer is fundamentally not trainable. However, some trainable pooling methods are proposed by researchers, e.g., Kobayashi 2019a.



(a) *Traditional Convolutional Network.* (b) *Swapped Pooling-Activation Network.*

Figure 6.29. *A traditional convolutional network (left) has a nonlinear activation, e.g., ReLU, before the pooling layer. The nonlinear activation distorts the distribution of the feature map outputted by the convolutional layer. Due to this, only a maximum summary statistic in MaxPool works in most problems. Swapping the order of pooling and activation (right) remedies the issue of distortion. In this network, the feature map's distribution is undisturbed which allows the use of a variety of summary statistics, including combinations of sufficient and ancillary statistics. For example, (mean, standard deviation), or (maximum, range).*

- **MLE statistic based on distribution**. As per Corollary 1 in § 6.11.3, the maximum likelihood estimator (MLE) makes the *best* statistic for pooling. Feature maps typically follow distributions from known families. If undisturbed, their MLEs can be used in pooling.
- **Usability of ancillary statistics**. A variety of ancillary statistics such as standard deviation, range, and IQR, become informative if the distribution is preserved. Moreover, a combination of MLE and ancillary statistics can also be pooled.

A swapped pooling and nonlinear activation architecture in the convolutional network allows the use of MLEs and ancillary statistics in pooling.

Both improvements have far-reaching effects. Especially, the possibility of pooling MLEs discussed in detail next.

6.13 Maximum Likelihood Estimators for Pooling

A feature map follows a distribution. The distribution differs with samples. For example, an object with sharp edges at the center of an image will have a different feature map distribution compared to an object with smudgy edges or located at a corner (shown in Figure 6.35a and 6.35b in § 6.14).

Regardless, the distribution's maximum likelihood estimator (MLE) makes the most efficient pooling statistic as described in § 6.11. A few distributions that feature maps typically follow and their MLEs are, therefore, given below.

Besides, a profound theoretical support for the parametric combination of the sample mean and maximum proposed by Boureau, Ponce, and LeCun 2010 as the optimal choice for pooling is given in § 6.13.4. The section shows that Boureau, Ponce, and LeCun 2010's parameterization is the MLE of a Weibull distribution.

6.13.1 Uniform Distribution

A uniform distribution belongs to the symmetric location probability distribution family. A uniform distribution describes a process where the random variable has an arbitrary outcome in a boundary denoted as (α, β) with the same probability. Its pdf is,

$$f(x) = \begin{cases} \dfrac{1}{\beta - \alpha} & \text{, if } \alpha < x < \beta \\ 0 & \text{, otherwise} \end{cases} \tag{6.13}$$

Different shapes of the uniform distribution are shown in Figure 6.30 as examples. Feature maps can follow a uniform distribution under some circumstances, such as if the object of interest is scattered in an image.

However, uniform distributions relevance lies in it being the maximum entropy probability distribution for a random variable. This implies, if nothing is known about the distribution except that the feature map is within a certain boundary (unknown limits) and it belongs to a certain class then the uniform distribution is an appropriate choice.

Besides, the maximum likelihood estimator of uniform distribution is,

$$\hat{\beta} = \max_i X_i \tag{6.14}$$

Therefore, if the feature map is uniformly distributed or distribution is unknown, $\max_i X_i$ is the *best* pooling statistic. The latter claim also reaffirms the reasoning behind max-pools superiority in § 6.12.1.

6.13.2 Normal Distribution

A normal distribution, a.k.a. Gaussian, is a continuous distribution from the exponential location family. It is characterized by its mean μ and standard deviation σ parameters. Its pdf is defined below and

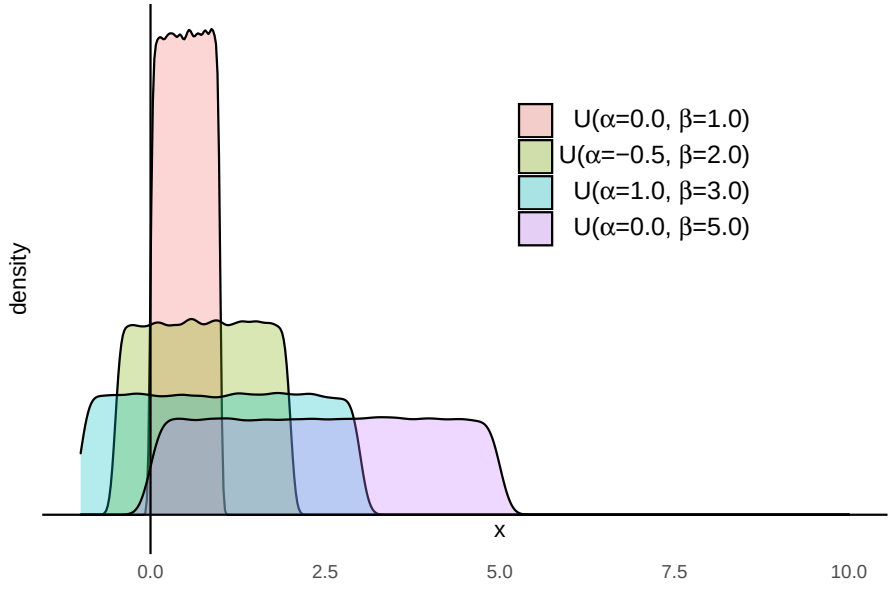


Figure 6.30. *In a uniform distribution, $X \sim U(\alpha, \beta)$, $X \in (\alpha, \beta)$, $\alpha, \beta \in \mathbb{R}$, and $\beta > \alpha$, the probability of the feature X having any value in (α, β) is equal $(= \frac{1}{\beta - \alpha})$ and zero, otherwise. It is also maximum entropy probability distribution, which implies if nothing is known about a feature map's distribution except that it is bounded and belongs to a certain class then uniform distribution is a reasonable default choice. The maximum likelihood estimate (MLE) of a uniform distribution is the maximum order statistic, i.e., $\hat{\beta} = X_{(n)} = \max_i X_i$.*

Figure 6.31 shows a few examples.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(- \frac{(x-\mu)^2}{2\sigma^2} \Big) \tag{6.15}$$

The MLEs of the normal distribution are

$$\hat{\mu} = \frac{\sum_i X_i}{n}, \tag{6.16}$$

$$\hat{\sigma}^2 = \frac{\sum_i (X_i - \bar{X})^2}{n-1}. \tag{6.17}$$

A normal distribution supports $-\infty < x < \infty$, i.e., $x \in \mathbb{R}$ and is symmetric. But most nonlinear activated feature map either distorts the symmetry or bounds it, e.g., ReLU lower bounds the feature map at 0.

Due to the distortion, activated feature maps are unlikely to follow a normal distribution. Therefore, a restructured *convolution* → *pooling* → *activation* architecture described in § 6.12.2 becomes favorable.

A normal distribution is a plausible distribution for most data samples barring some special cases such as if the object in an image is at the corners. Besides, normality offers two MLE statistics for pooling that provide both signal and spread information.

6.13.3 Gamma Distribution

A gamma distribution is an asymmetrical distribution also from the exponential family. It has two parameters: shape k and scale θ. They characterize the lopsidedness and spread of the distribution.

Its pdf is,

$$f(x) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} \exp - \frac{x}{\theta} \tag{6.18}$$

where, $x > 0$, and $k, \theta > 0$.

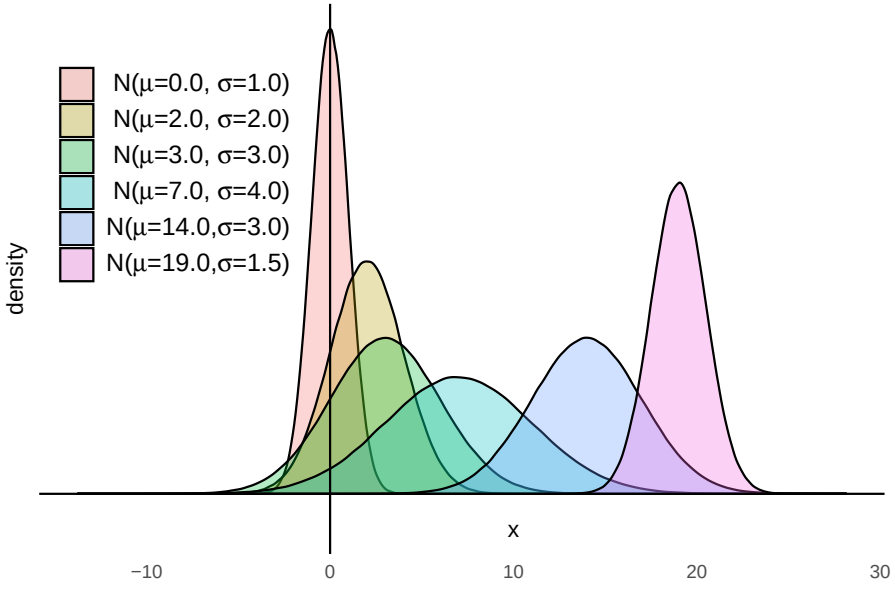


Figure 6.31. *A normal distribution, $X \sim N(\mu, \sigma)$, $X \in \mathbb{R}$, $\mu \in \mathbb{R}, \sigma \in \mathbb{R}^+$, is the symmetric distribution from the exponential family. A convolutional feature map follows a normal distribution if and only if it can take any value in $(-\infty, \infty)$ and the probability $P(X = x)$ is symmetric. That is, a nonlinear activated convolutional output is unlikely to be normal. A linearly activated (or inactivated) convolutional output can be assumed to be normal in many data sets. The center μ and the spread σ differs by the data. The MLEs for them are, $\hat{\mu} = \bar{X} = \frac{\sum_i X_i}{n}$ and $\hat{\sigma}^2 = S^2 = \frac{\sum_i (X_i - \bar{X})^2}{(n-1)}$.*

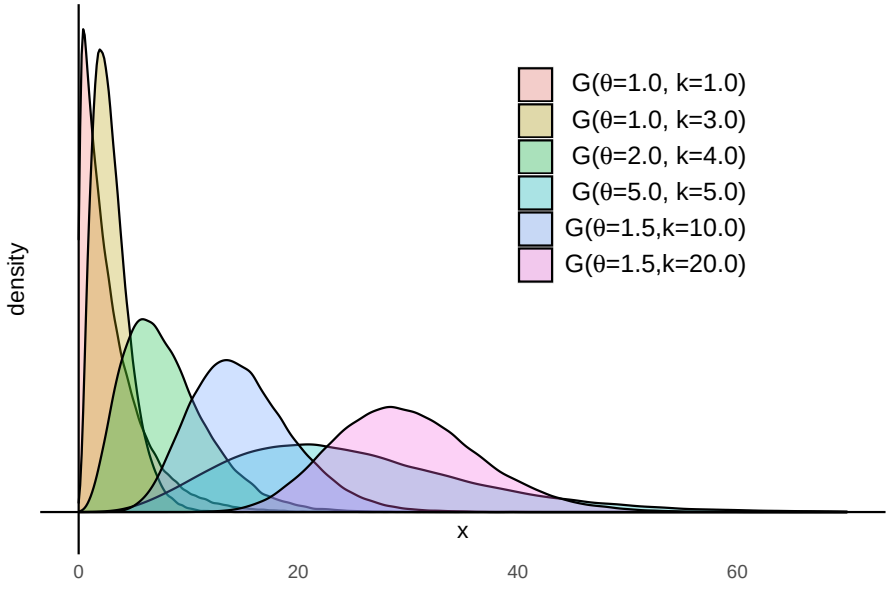


Figure 6.32. *A gamma distribution, $X \sim G(k, \theta)$, $X > 0$, and $k, \theta > 0$, is one of the asymmetric distribution from the exponential family. Gamma distribution is applicable on an activated convolutional output x such that $f(x) > 0$ or constrained convolution where $x > 0$. Gamma distribution provides more flexibility due to its asymmetry. The exponential distribution is a special case of Gamma distribution if $k = 1$, shown with $G(1, 1)$ above. This is an extreme case found when the object is partially present or at corners of the input. In other cases, the shape and scale differ with data and are characterized by k and θ, respectively. Closed-form MLEs do not exist for Gamma, however, mixed type log-moment estimators exist that have similar efficiency as MLEs presented as,*

$$\hat{\theta} = \frac{1}{n(n-1)}\left(n \sum_i x_i \log(x_i) - \sum_i \log(x_i) \sum_i x_i\right) \text{ and}$$

$$\hat{k} = \frac{n \sum_i x_i}{n \sum_i x_i \log(x_i) - \sum_i \log(x_i) \sum_i x_i}.$$

The distribution can take a variety of forms as shown in Figure 6.32. In fact, like uniform distribution, the gamma distribution is also the maximum entropy probability distribution. This makes gamma distribution applicable to feature maps in most data samples.

However, its support is $x > 0$ which constrains its applicability only to positive-valued feature maps. There are several ways to constrain the feature maps, e.g., using a positive activation such as ReLU (or Null-ReLU defined in Appendix G), or with kernel and bias constraints set to non-negative in the convolutional layer and non-negative input[17]. Besides, if needed, a three-parameter Gamma distribution is also available with a location parameter c and pdf $f(x|c,\theta,k) = \frac{1}{\Gamma(k)\theta^k}(x-c)^{k-1}\exp -\frac{(x-c)}{\theta}$ where $c \in (-\infty, \infty)$.

Assuming the feature map is positive, i.e., $c = 0$, gamma distribution applies to them in most cases. If $k = 1$, $G(\theta, 1)$ is an exponential distribution also shown in Figure 6.32. Another example in Figure 6.35b in § 6.14 shows an image with an object at a corner yields an exponentially distributed feature map. The MLE for exponential is the sample mean,

$$\hat{\theta} = \frac{\sum_i x_i}{n}. \tag{6.19}$$

Under other circumstances, the gamma distribution is flexible to take different shapes. However, a closed-form MLE does not exist for them. Fortunately, mixed type log-moment estimators exist that have similar efficiency as MLEs (Ye and N. Chen 2017). They are,

$$\hat{\theta} = \frac{1}{n^2}\left(n\sum_i x_i \log(x_i) - \sum_i \log(x_i)\sum_i x_i\right) \tag{6.20}$$

$$\hat{k} = \frac{n\sum_i x_i}{n\sum_i x_i \log(x_i) - \sum_i \log(x_i)\sum_i x_i}. \tag{6.21}$$

[17] E.g. Input scaled with `sklearn.preprocessing.MinMaxScaler()` and Convolutional layer defined as
`Conv1D(..., kernel_constraint=tf.keras.constraints.NonNeg(), bias_constraint=tf.keras.constraints.NonNeg(),...)`.

Based on the listed complete sufficient statistic for the gamma distribution in Table 6.2 as per Theorem 9, the estimators $\hat{\theta}$ and $\hat{k}$ are complete sufficient statistics. However, the statistics are biased. The bias-corrected statistics are (Louzada, P. L. Ramos, and E. Ramos 2019),

$$\tilde{\theta} = \frac{n}{n-1}\hat{\theta} \tag{6.22}$$

$$\tilde{k} = \hat{k} - \frac{1}{n}\left(3\hat{k} - \frac{2}{3}\left(\frac{\hat{k}}{1+\hat{k}}\right) - \frac{4}{5}\frac{\hat{k}}{(1+\hat{k})^2}\right). \tag{6.23}$$

These unbiased estimators are a function of the complete sufficient statistics. Therefore, according to Theorem 6 $\tilde{\theta}$ and $\tilde{k}$ are MVUE. In effect, they exhibit the same properties expected from MLEs for pooling.

Note that, in practice, $\log(x)$ in the statistics can be replaced with $\log(x+\epsilon)$ where ϵ is a small constant in $\mathbb{R}^+$, e.g., $\epsilon \leftarrow 1e-3$. This adds a small bias to the estimates but also makes them stable. This correction addresses the possibility of feature maps taking extremely small values.

6.13.4 Weibull Distribution

A Weibull distribution is also an asymmetrical distribution from the exponential family. It has similar parameters as a gamma distribution, scale λ and shape k, and also a similar form. Like a gamma distribution, Weibull is also lopsided and has a spread characterized by k and λ. But they differ in their precipitousness. This is clearer from the pdf given as,

$$f(x) = \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} \exp -\left(\frac{x}{\lambda}\right)^{k} \tag{6.24}$$

where, $x \geq 0$, and $\lambda, k > 0$.

Ignoring the constants in the pdf of gamma and Weibull distributions (in Equation 6.18 and Equation 6.24, respectively), one can note that the difference is $\exp(-x/\theta)$ versus $\exp -(x/\lambda)^k$. This implies that Weibull distribution precipitates rapidly if $k > 1$ and slowly if $k < 1$, while the

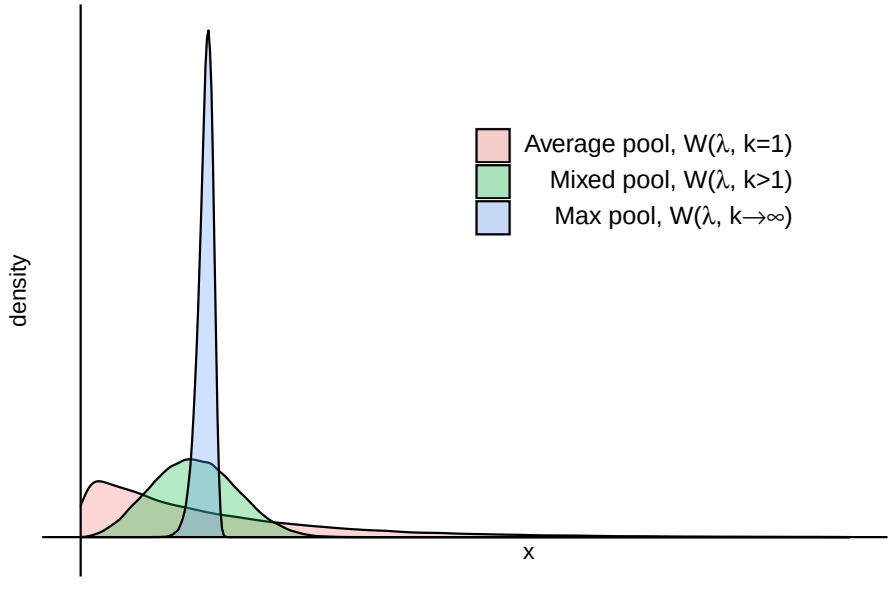


Figure 6.33. *Pooling has typically stayed with max or average statistics. A better pooling, however, is hypothesized to be somewhere between them. Boureau, Ponce, and LeCun 2010 parameterized pooling statistic as* $f(x) = \left(\frac{1}{n}\sum_i x_i^k\right)^{\frac{1}{k}}$*, which gives the average for* $k = 1$ *and the max for* $k \to \infty$*. A value in between for* k *gives a pooling statistic that is a "mixture" of average and max. This parameterization comes from the assumption that* X *is Weibull distributed, i.e.,* $X \sim W(\lambda, k)$*, and the pooling statistic is the MLE for* λ*.*

gamma distribution is indifferent. If $k = 1$, both gamma and Weibull are equal and reduces to an exponential distribution.

The ability to adjust the precipitousness makes Weibull distribution a sound choice for fitting feature maps. Because the separability of features defines the precipitousness. For example, a strong feature map sharply distinguishes an object's features and will have a quickly dropping off pdf, i.e., a large k.

Therefore, the MLE of Weibull distribution given below is quite appropriate for pooling.

$$\hat{\lambda} = \left(\frac{1}{n} \sum_i x_i^k \right)^{\frac{1}{k}} \tag{6.25}$$

assuming k is known.

Interestingly, the parametrization given in Boureau, Ponce, and LeCun 2010 is the MLE of Weibull distribution. This is interesting because in their or other prior work a connection between pooling and Weibull distribution was not established.

Instead, Boureau, Ponce, and LeCun 2010 found that the optimal pooling statistic is somewhere between an average- and max-pooling. They, therefore, posed the pooling statistic as in Equation 6.25 that continuously transitions from average- to max-pooling as follows,

- if $k = 1$, then average-pooling,
- if $k \to \infty$, then max-pooling, and
- if $1 < k < \infty$, then mixed-pooling.

The Weibull distribution in these conditions are shown in Figure 6.33. In the figure, it is noticed that under the max-pool condition the distribution precipitates quickly indicating a distinguishing feature map. The average-pool is for $k = 1$ when Weibull becomes an Exponential distribution for which the MLE is indeed the sample mean (Equation 6.19). A mixed pooling is in between the two and so is the shape of its distribution.

In the above, the shape parameter k is assumed to be known. Fitting a Weibull distribution to feature maps can become more informative if the k is also derived from the features (data). The distribution's form for different (λ, k) are shown in Figure 6.34 for illustration.

Unfortunately, a closed-form MLE for k does not exist. It can be estimated using numerical methods by solving,

$$\frac{\sum_i x_i^k \log(x_i)}{\sum_i x_i^k} - \frac{1}{k} - \frac{1}{n}\sum_i \log(x_i) = 0 \tag{6.26}$$

Moreover, Weibull distribution can also be fitted if feature maps are lower bounded at τ, where $\tau > 0$. Then the MLE for $\hat{\lambda}$ is,

$$\hat{\lambda} = \left(\frac{1}{\sum_i \mathbb{1}(x_i > \tau)} \sum_i (x_i^k - \tau^k)\mathbb{1}(x_i > \tau)\right)^{\frac{1}{k}} \tag{6.27}$$

and k can be estimated by solving,

$$\frac{\sum_i (x_i^k \log(x_i) - \tau^k \log(\tau))}{\sum_i (x_i^k - \tau^k)} - \frac{1}{\sum_i \mathbb{1}(x_i > \tau)} \sum_i \log(x_i)\mathbb{1}(x_i > \tau) = 0 \tag{6.28}$$

where $\mathbb{1}(x_i > \tau)$ is an indicator function equal to 1, if $x_i > \tau$, and 0, otherwise.

These expressions are derived from a three-parameter Weibull defined as, $f(x) = \frac{k}{\lambda}\left(\frac{x-\tau}{\lambda}\right)^{k-1} \exp -\left(\frac{x-\tau}{\lambda}\right)^k$, where $\tau < x < \infty$ (R. L. Smith 1985; Hirose 1996)[18]. They are helpful to accommodate some types of activation or dropout techniques.

Lastly, like gamma, Weibull distribution also supports only positive x. Therefore, approaches, such as positive-constrained inputs, kernel, and bias, discussed in § 6.13.3 can be used.

[18] A similar three-parameter is also for gamma distribution in R. L. Smith 1985 but closed-form estimators are unavailable.

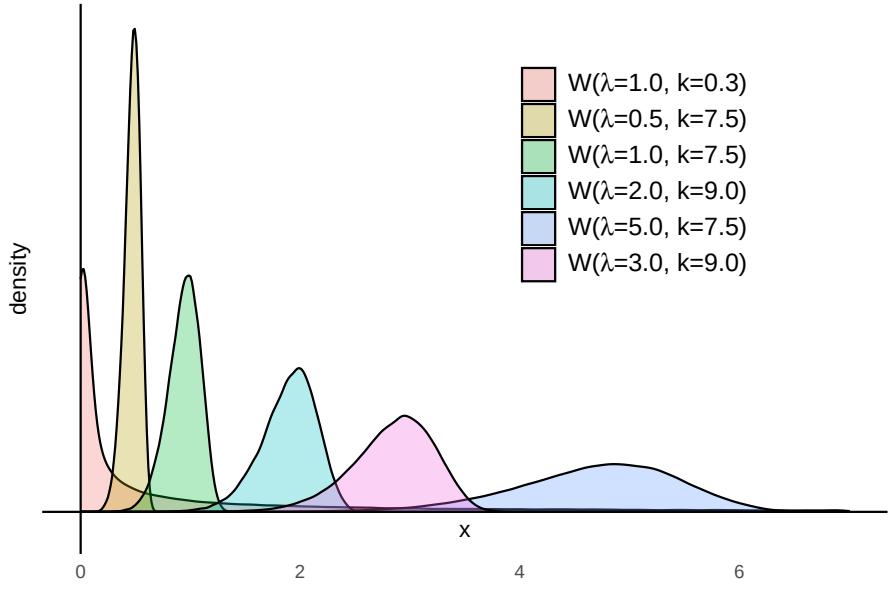


Figure 6.34. *Weibull distribution, $X \sim W(\lambda, k)$, $X \in \mathbb{R}^+$, and $\lambda, k \in \mathbb{R}^+$, is another asymmetric distribution from the exponential family. Similar to a Gamma distribution, Weibull provides flexibility for different shapes and scales. Moreover, the Exponential distribution is also a special case of Weibull if $k = 1$. MLE is available for λ in Weibull, $\hat{\lambda} = \left(\frac{1}{n}\sum_i x_i^k\right)^{\frac{1}{k}}$. Closed-form MLE for k is, however, unavailable and can be estimated via numerical methods.*

.

6.14 Advanced Pooling

The possibility of fitting distributions uncovered a myriad of pooling statistics. It also made possible using advanced techniques, such as an adaptive selection of distribution.

Such techniques have significance because optimal pooling depends on the characteristics of feature maps in convolutional networks and the data set. Automatically determining the optimal pooling is challenging and discussed in expanse in Lee, Gallagher, and Tu 2016; Kobayashi 2019b.

Moreover, MLEs are learned to be the appropriate pooling statistic. But they are unavailable for some distributions.

Besides, convolutional feature maps are likely to be dependent in most data sets because nearby features are correlated. However, pooling statistics are developed assuming their independence.

This section provides a few directions to address these challenges.

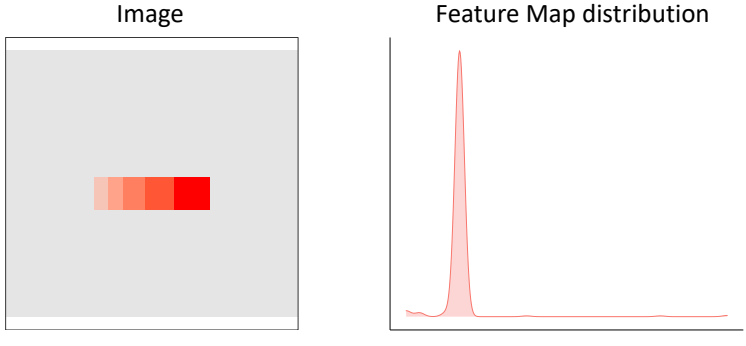
6.14.1 Adaptive Distribution Selection

Feature maps for different samples in a data set can differ significantly. For illustration, Figure 6.35a and 6.35b show images with an object at the center and corner, respectively. The images are filtered through a Sobel filter (§ 12.6.2 in McReynolds and Blythe 2005). Also shown in the figures is the feature map distribution yielded by the filter.

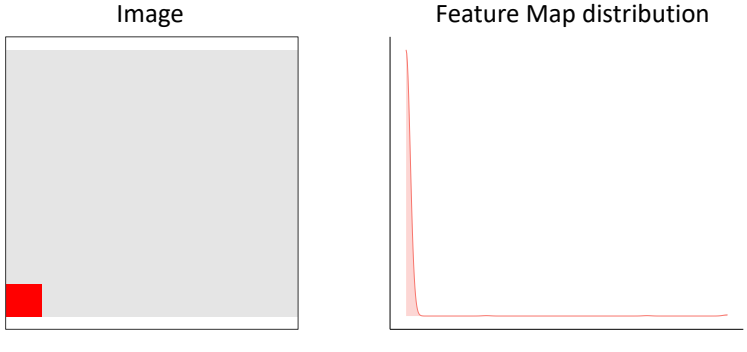
The former image results in a peaked distribution that can be from a normal, gamma, or Weibull while the latter results in an exponential-like distribution.

If the distribution is known, using MLEs for pooling is fitting the distribution to the feature map.

This is straightforward with normal and gamma distribution as closed-form estimators exist for their parameters. For Weibull, MLE is available for scale λ but not for the shape k. Although k can be numerically estimated by solving Equation 6.26 for k, it is computationally intensive. However, k need not be estimated with precision. Instead, k can be assumed to belong in a finite discrete set of positive real numbers, i.e.,



(a) *Object with color gradient.*



(b) *Object on a corner.*

Figure 6.35. *An image containing an object with a color gradient (top) and another image with an object at a corner (bottom) are convolved with a Sobel kernel that detects the color gradient changes. The outputted convolution feature map has a gamma (top) and exponential (bottom) distribution, respectively. Their respective MLEs should be used in pooling.*

$\mathcal{K} = \{k | k \in \mathbb{R}^+\}$ and $|\mathcal{K}|$ is small. The $k \in \mathcal{K}$ that yields the maximum likelihood should be chosen, i.e.,

$$\arg_k \max \prod_i f(x_i | \hat{\lambda}, k) \tag{6.29}$$

where f is the pdf of Weibull in Equation 6.24, $\hat{\lambda}$ is estimated using Equation 6.25, and $x_i, i = 1, \ldots, n$ is the feature map.

Deriving pooling statistics by fitting distributions is effective. But it works on an assumption that the distribution is known. Since, exponential distributions, viz. normal, gamma, and Weibull, provide reasonable flexibility, the assumption is not strong. Still further improvement can be made by adaptively choosing a distribution, i.e., let the distribution be data-driven. An approach could be to fit various distributions and select the one which yields the maximum likelihood.

6.14.2 Complete Statistics for Exponential Family

We have learned that MLEs are the best pooling statistic. But their closed-form expressions are sometimes unknown.

We know that MLEs are a function of complete statistic(s). In absence of an MLE expression, complete statistic(s) can be used in pooling.

Most feature maps follow a distribution from exponential family and, fortunately, complete statistics for any distribution from the family is available based on Theorem 9 below.

Theorem 9. ***Exponential family complete sufficient statistics***[19]. *Let $X_1, \ldots, X_n$ be iid observations from an exponential family with pdf or pmf of form*

$$f(x|\boldsymbol{\theta}) = h(x)c(\boldsymbol{\theta}) \exp\left(\sum_{j=1}^{k} w_j(\boldsymbol{\theta}) t_j(x)\right) \tag{6.30}$$

[19] Based on Theorem 8.1 in Lehmann and Scheffé 1955 and Theorem 6.2.10 in Casella and Berger 2002.

Table 6.2. List of Complete Sufficient Statistics based on Theorem 9 for Exponential family distributions.

Distribution	**Pdf**, x, $\boldsymbol{\theta}$	**Complete sufficient statistics**, $T(\boldsymbol{X})$
Normal	$\frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{(x-\mu)^2}{2\sigma^2}$, $x \in (-\infty, \infty)$, $\boldsymbol{\theta} = (\mu, \sigma^2)$	$\left(\sum_{i=1}^n X_i, \sum_{i=1}^n X_i^2\right)$
Exponential	$\lambda \exp(-\lambda x)$, $x \in [0, \infty)$, $\boldsymbol{\theta} = \lambda$	$\sum_{i=1}^n X_i$
Gamma	$\frac{\beta^\alpha}{\Gamma(\alpha)}\exp -\Big(\beta x - (\alpha-1)\log x\Big)$, $x \in (0, \infty)$, $\boldsymbol{\theta} = (\alpha, \beta)$	$\left(\sum_{i=1}^n X_i, \sum_{i=1}^n \log X_i\right)$
Weibull	$\frac{k}{\lambda}\exp -\Big(\big(\frac{x}{\lambda}\big)^k - (k-1)\log\frac{x}{\lambda}\Big)$, $x \in [0, \infty)$, $\boldsymbol{\theta} = (\lambda, k)$	$\left(\sum_{i=1}^n X_i^k, \sum_{i=1}^n \log X_i\right)$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$. *Then the statistic*

$$T(\boldsymbol{X}) = \left(\sum_{i=1}^n t_1(X_i), \sum_{i=1}^n t_2(X_i), \ldots, \sum_{i=1}^n t_k(X_i)\right) \tag{6.31}$$

is complete if $\{(w_1(\boldsymbol{\theta}), \ldots, w_k(\boldsymbol{\theta})) : \boldsymbol{\theta} \in \Theta\}$ *contains an open set in* $\mathbb{R}^k$. *Moreover, it is also a sufficient statistic for* θ.

It can be noted that the MLEs for normal, gamma and Weibull distributions are indeed a function of the complete statistics listed in

Table 6.2 based on the theorem. Similarly, complete statistic(s) for any other distribution from the exponential family can be determined for pooling.

6.14.3 Multivariate Distribution

Most of the pooling approaches assume feature maps are independent. The independence assumption is, however, false since close by image features are correlated. For example, if a filter detects an edge then it is likely to find the next pixel as an edge as well. Consequently, the feature map it yields will have dependence.

Addressing the dependence is challenging with traditional pooling methods. Only a few techniques address it. Saeedan et al. 2018 is one of them which uses the concept developed in Weber et al. 2016 in the field of image processing for detail-preservation for downscaling in pooling.

The dependence can be addressed by making the features independent, e.g., removing autocorrelation, before fitting distributions (pooling). Features dependence can also be addressed by fitting multivariate distributions, e.g., multivariate normal and Wishart distributions.

6.15 History of Pooling

Pooling is an important construct of a convolutional network. Without pooling, the network is statistically and computationally inefficient, and virtually dysfunctional.

The concept has roots in biological constructs. Many computer vision architectures including pooling in convolutional networks have been inspired by studies of the visual cortex on multi-stage processing of simple and complex cells in Hubel and Wiesel 1962. Translational invariance is achieved by the complex cells that aggregate local features in a neighborhood.

One of the earliest neural networks with the pooling concept is the Neocognitron (Fukushima 1986). This network employed a combination of “S-cells” and “C-cells” which acted as activation and pooling, respec-

tively. The "C-cells" become active if at least one of the inputs from the "S-cells" is active. This is similar to a binary gate that makes the network robust to slight deformations and transformations.

LeCun, Boser, et al. 1990 introduced the idea of parameter sharing with convolution and network invariance via subsampling by taking an average of the local features. Average-pooling was further used in LeCun, Bottou, et al. 1998.

Max-pooling was put forward soon after in Riesenhuber and Poggio 1999. The paper discusses the biological functioning of the visual cortex and lays two idealized pooling mechanisms, linear summation ('SUM') with equal weights (to achieve an isotropic response), and a nonlinear maximum operation ('MAX'), where the strongest afferent determines the postsynaptic response. They are average and max-pooling, respectively.

Riesenhuber and Poggio 1999 compared average- and max- pooling from a biological visual cortex functioning standpoint. They explained that responses of a complex cell would be invariant as long as the stimulus stayed in the cell's receptive field. However, it might fail to infer whether there truly is a preferred feature somewhere in the complex cell's receptive field. In effect, the feature specificity is lost. However, in max-pooling the output is the most active afferent and, therefore, signals the best match of the stimulus to the afferent's preferred feature. This premise in Riesenhuber and Poggio 1999 explained the reason behind max-pool's robustness over the average.

Max-pool was further used and empirical evidence of its efficacy was found in Gawne and Martin 2002; Lampl et al. 2004; Serre, Wolf, and Poggio 2005; Ranzato, Boureau, and Cun 2008. Using max-pool, Yang et al. 2009 reported much better classification performance on multi-object or scene-classification benchmarks compared to average-pool. However, no theoretical justification behind max-pool's outperformance was yet given.

Boureau, Ponce, and LeCun 2010 perhaps provided the earliest theoretical support behind max-pool. They assumed feature maps as Bernoulli random variables that take values 0 or 1. Under the assumption, they expressed the mean of separation and variance of max-pooled features.

Their expressions show that max-pool does better class separation than average. However, the justification was based on an extreme simplification of Bernoulli distribution while feature maps are continuous in most problems. To which, Ranjan 2020 recently provided more general proof from a statistical standpoint.

Besides, the possibility of the optimal pooling lying in between average- and max- pooling was seeded in Boureau, Ponce, and LeCun 2010. They, themselves, also provided a parameterization to combine both as $\sum_i \frac{\exp(\beta \boldsymbol{x}_i + \alpha)}{\sum_j \exp(\beta \boldsymbol{x}_j + \alpha)}$ which is equivalent to average or max if $\beta \to 0$ and $\beta \to \infty$, respectively, and $\alpha = 0$. A more sophisticated approach to estimate the α, β from features, i.e., trainable mixing parameters, based on maximum entropy principle was developed in Kobayashi 2019b.

The mixing idea was taken in D. Yu et al. 2014; Lee, Gallagher, and Tu 2016 to propose mixed-pooling as $f_{mix}(\boldsymbol{x}) = \alpha \cdot f_{max}(\boldsymbol{x}) + (1 - \alpha) \cdot f_{avg}(\boldsymbol{x})$. Lee, Gallagher, and Tu 2016 also devised a trainable mixing called as gated-pooling in which the weight $\alpha = \sigma(\boldsymbol{\omega}^T \boldsymbol{x})$ where σ is a sigmoid function in $[0, 1]$ and $\boldsymbol{\omega}$ is learned during training. A further extension, called Tree-pooling, was also proposed by them that automatically learned the pooling filters and mixing weights to responsively combine the learned filters.

Although pooling is typically used for invariance, a variant called stochastic pooling was proposed in Zeiler and Fergus 2013 for regularization of convolutional networks. Stochastic pooling works by fitting a multinomial distribution to the features and randomly drawing a sample from it.

Most of the pooling techniques worked by assuming the independence of features. However, it is a strong assumption because nearby features are usually correlated. Ranjan 2020 discussed this issue and provided a few directions, e.g., fitting a multivariate distribution and using its summary statistic or removing feature dependency before computing a statistic for pooling. Prior to this, T. Williams and Li 2018 and Saeedan et al. 2018 have worked in a similar direction.

T. Williams and Li 2018 proposed a wavelet pooling that uses a second-level wavelet decomposition to subsample features. This ap-

proach was claimed to resolve the shortcomings of the nearest neighbor interpolation-like method, i.e., local neighborhood features pooling, such as edge halos, blurring, and aliasing Parker, Kenyon, and Troxel 1983.

Saeedan et al. 2018 developed a "detail-preserving" pooling drawn from the approach for image processing in Weber et al. 2016 called detail-preserving image downscaling (DPID). DPID calculates a weighted average of the input, but unlike traditional bilateral filters (Tomasi and Manduchi 1998), it rewards the pixels that have a larger difference to the downscaled image. This provides a customizable level of detail magnification by allowing modulation of the influence of regions around edges or corners.

Kobayashi 2019a proposed fitting Gaussian distribution on the local activations and aggregate them into sample mean μ_X and standard deviation σ_X. They devised the pooling statistic as $\mu_X + \eta\sigma_X$ where $\mu_X = \frac{\sum_i X_i}{n}$ and $\sigma_X = \sqrt{\frac{\sum_i (X_i - \bar{X})^2}{(n-1)}}$. Since the activations are not full Gaussian, they fit half-Gaussian and inverse softplus-Gaussian. Still, the activations are unlikely to follow these distributions due to the phenomenon of distribution distortion presented in Ranjan 2020. Moreover, the $\mu_X + \eta\sigma_X$ is not a *complete* statistic due which it does not have the minimum variance.

Besides, He et al. 2015b developed a spatial pyramid pooling. It is mainly designed to deal with images of varying size, rather than delving in to different pooling functions or incorporating responsiveness.

There is another school of thought that rallies against pooling altogether. For example, Springenberg et al. 2015 proposed an "all Convolutional Net" which replaced the pooling layers with repeated convolutional layers. To reduce the feature map size, they used larger stride in some of the convolutional layers to achieve a similar performance as with the pooling layers. Variational autoencoders (VAEs) or generative adversarial networks (GANs) are also found to work better in absence of pooling layers. However, this could be due to the usage of lossy statistics[20] for pooling. A use of MLEs or complete sufficient statistics as proposed in Ranjan 2020 would work well in VAEs or GANs.

[20]Summary statistics that lose the information of the original data.

6.16 Rules-of-thumb

- **Baseline network**. Construct a simple sequential baseline model with *convolution* → *pooling* → *activation* → *flatten* → *dense* → *output* layer structure. Note to swap activation and pooling layers.

- **Convolution layer**
  - `Conv1D` **vs.** `Conv2D` **vs.** `Conv3D`. A `Conv'x'D` is chosen based on the number of spatial axes in the input. Use `Conv1D`, `Conv2D`, and `Conv3D` for inputs with 1, 2, and 3 spatial axes, respectively. Follow Table 6.1 for details.
  - **Kernel**. The `kernel_size` argument is a tuple in which each element represents the size of the kernel along a spatial axis. The size can be taken as the $\sqrt{\text{spatial axis size}}/2$.
  - **Filters**. The number of `filters` can be taken as a number from the geometric series of 2 close to n_features/4.
  - **Activation**. Use a `linear` activation. A nonlinear activation will be added after the pooling layer.
  - **Padding**. Use `valid` padding in a shallow network. A shallow network is defined as one in which the feature map size does not reduce significantly before the output layer. In deep networks with several convolutional layers, use the `same` padding at least in some of the convolutional layers.
  - **Dilation**. Do not use dilation in a baseline model. In deep networks, undilated and dilated convolutional layers can be paired.
  - **Stride**. Use the default `stride=1` in a baseline model. The stride can be increased to `2` if the input dimension is high. However, in most problems, a stride larger than `2` is not recommended.

- **Pooling layer**
  - `Pool1D` **vs.** `Pool2D` **vs.** `Pool3D`. Use the pooling layer consistent with the `Conv` layer. For example, `Pool1D` with `Conv1D`, and so on.