Parameter Estimation, Covariance, KL Divergence, and Softmax

Gregory Glickert

February 20, 2026

Contents

1 Maximum Likelihood Estimation (MLE)

MLE estimates distribution parameters by finding the values that maximize the probability of observing the data. For i.i.d. samples $x_1, \ldots, x_n$ with density $f(x|\theta)$:

$$\hat{\theta} = \arg\max_{\theta} \sum_{i=1}^{n} \log f(x_i \mid \theta)$$

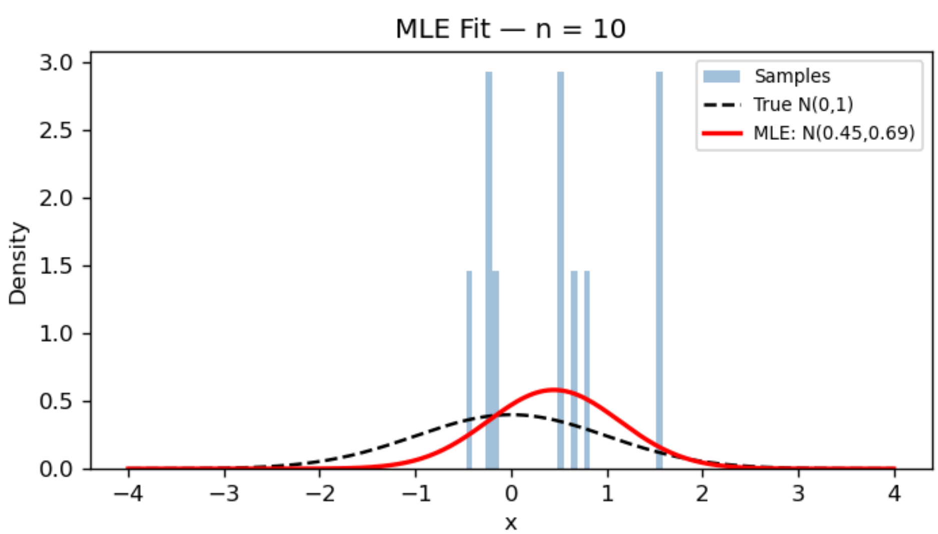
Working with the log converts the product into a sum, which is numerically stable and analytically convenient.

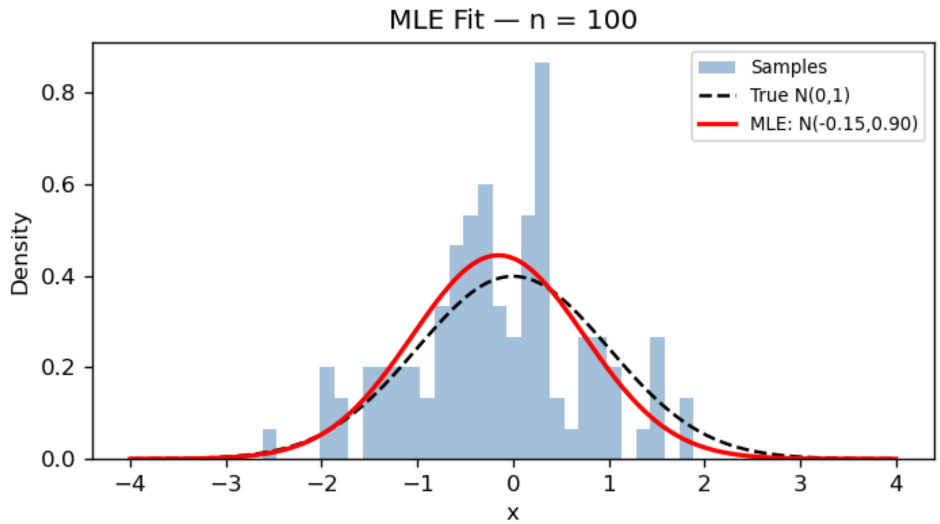
For a normal distribution $N(\mu, \sigma^2)$, differentiating the log-likelihood and setting it to zero gives closed-form estimators:

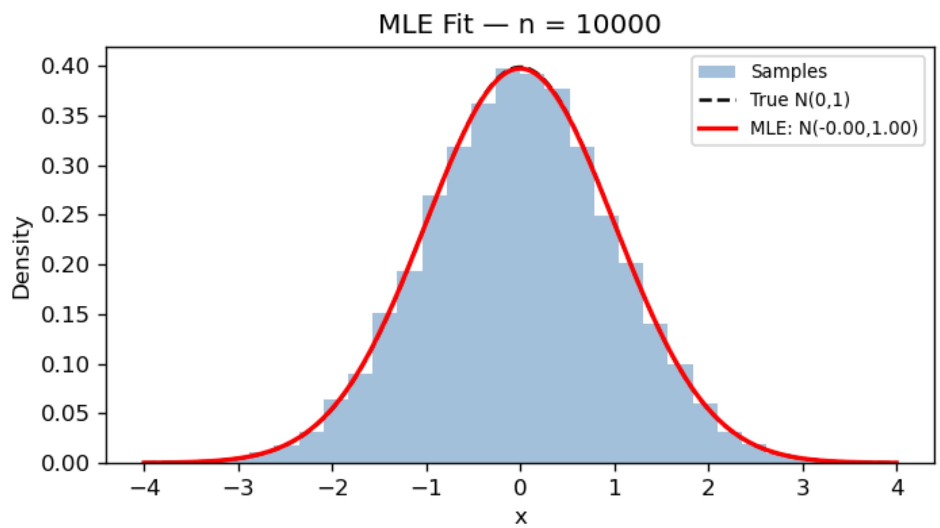
$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

Effect of Sample Size

With more data the MLE estimate converges to the true parameters. The three plots below show fits to samples of size $n = 10$, 100, and 10000 drawn from $N(0, 1)$:







At $n = 10$ the fit can be noticeably off; by $n = 10000$ the estimated curve is virtually indistinguishable from the true distribution.

2 Method of Moments (MoM)

Method of Moments estimates parameters by equating theoretical moments to sample moments. The k-th theoretical moment is $\mu_k(\theta) = \mathbb{E}[X^k]$; we solve for θ such that $\mu_k(\hat{\theta}) = \frac{1}{n}\sum x_i^k$.

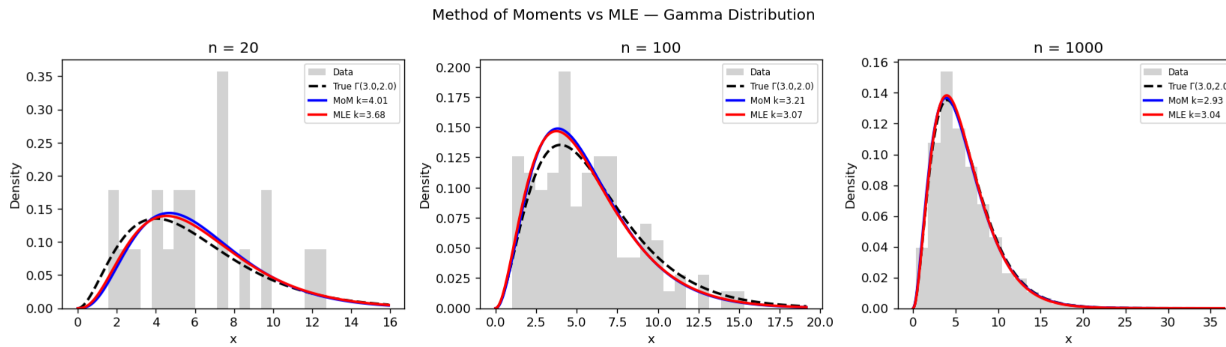
For a Gamma distribution Gamma(k, θ) with mean $k\theta$ and variance $k\theta^2$, matching the first two moments gives:

$$\hat{\theta}_{\text{MoM}} = \frac{\hat{\sigma}^2}{\bar{x}} \qquad \hat{k}_{\text{MoM}} = \frac{\bar{x}^2}{\hat{\sigma}^2}$$

MoM vs. MLE: When to Use Each

	Method of Moments	MLE
Approach	Match sample moments to theory	Maximise likelihood of data
Computation	Closed-form; very fast	May require numerical optimisation
Efficiency	Can be less statistically efficient	Asymptotically optimal
Use when	Quick estimate; complex likelihood	Sufficient data; accuracy matters

The figure below compares both methods on Gamma-distributed data. At small n they can diverge; at large n both converge to the true distribution.

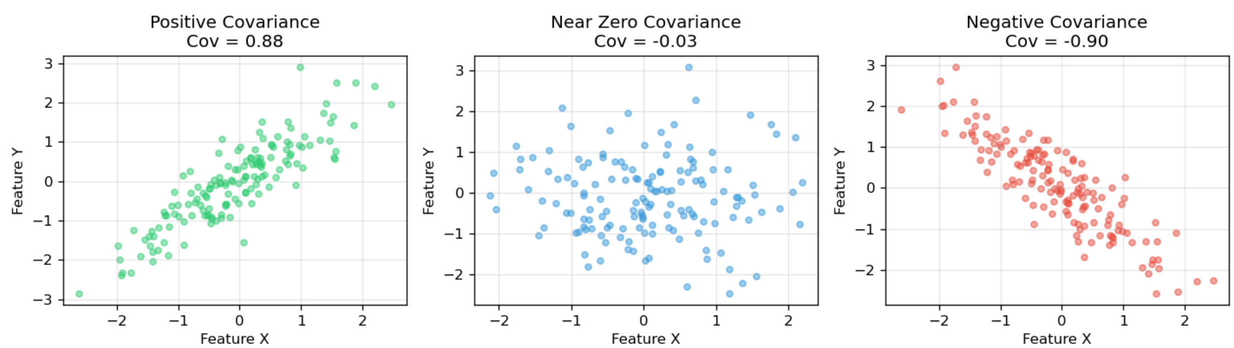


3 Covariance and Feature Selection

Covariance measures how two features vary together:

$$\mathrm{Cov}(X,Y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

Positive values mean features increase together; negative means they move inversely; near zero means they are uncorrelated.

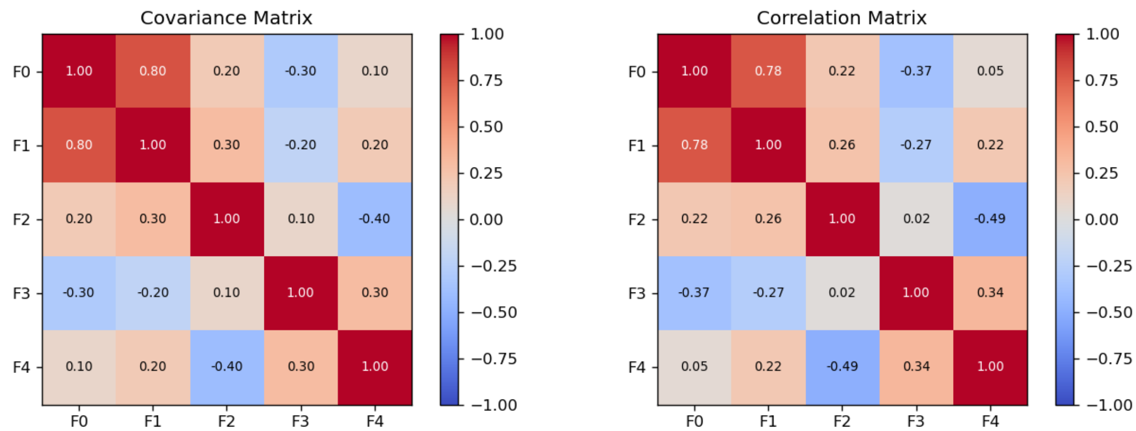


Covariance and Correlation Matrices

For d features, all pairwise covariances form a $d \times d$ covariance matrix Σ. Normalising each entry to $[-1, 1]$ gives the correlation matrix:

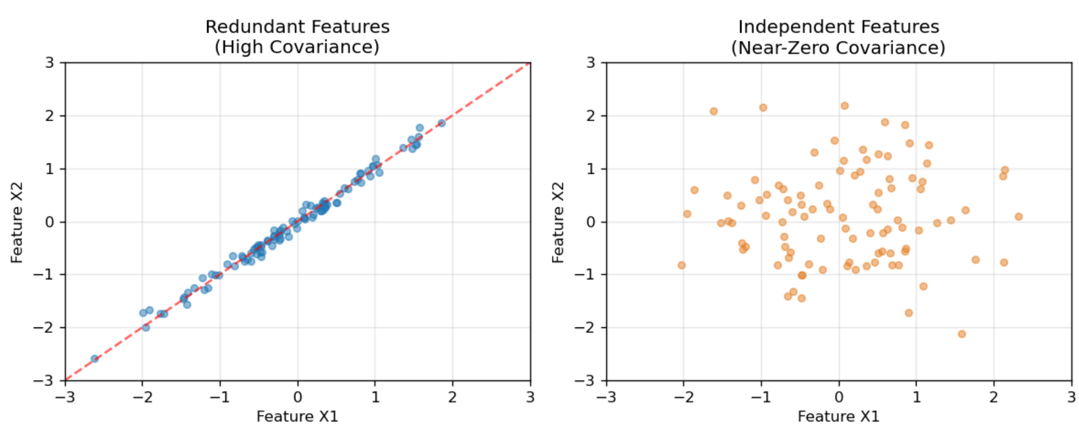
$$\rho_{ij} = \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathrm{Var}(X_i)\,\mathrm{Var}(X_j)}}$$

High off-diagonal values flag redundant features.



Why It Matters for Machine Learning

- **Redundancy detection.** Highly correlated features carry overlapping information; dropping one reduces cost with no information loss.
- **Multicollinearity.** Correlated features destabilise linear models and inflate coefficient variance.
- **Dimensionality reduction.** PCA uses the covariance matrix to find directions of maximum variance, compressing d features into far fewer dimensions.
- **Interpretability.** Independent features have clearer, separable effects on model predictions.



4 KL Divergence

KL divergence quantifies how different one distribution is from another:

$$D_{KL}(P\|Q) = \int p(x) \log \frac{p(x)}{q(x)} \, dx$$

It is *not* symmetric: $D_{KL}(P\|Q) \neq D_{KL}(Q\|P)$. For two normals with equal variance:

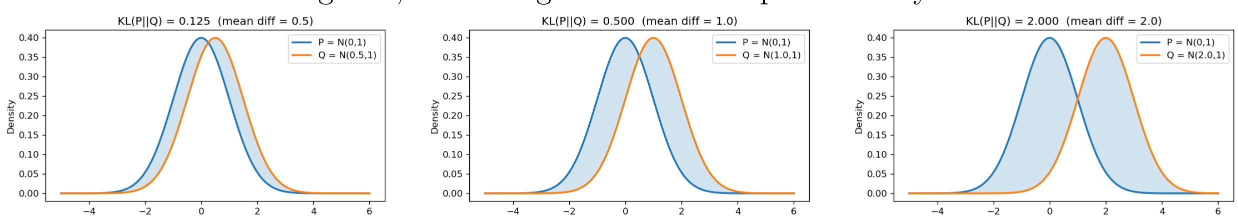
$$D_{KL}\big(N(\mu_1, \sigma^2) \,\|\, N(\mu_2, \sigma^2)\big) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2}$$

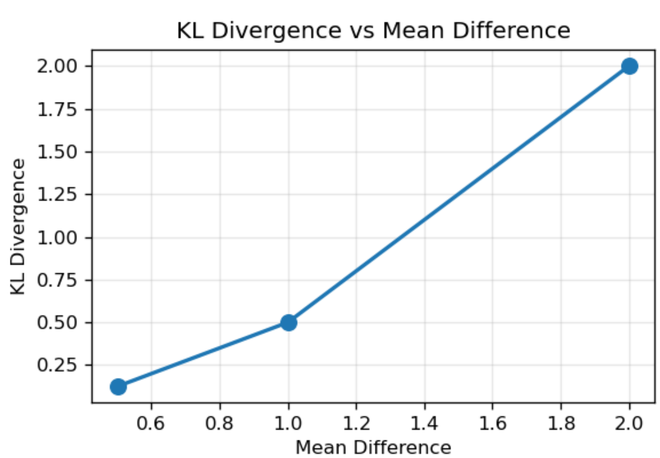
Applications in Machine Learning

- **Model selection.** Measures how well an approximate distribution fits the true one.
- **Variational inference.** Minimises KL between an approximate and the true posterior.
- **GANs.** Related divergences guide generator training by comparing real and generated data distributions.
- **Cross-entropy loss.** Minimising cross-entropy is equivalent to minimising KL from data to model predictions.

Visualisation

As the mean difference grows, KL divergence increases quadratically.





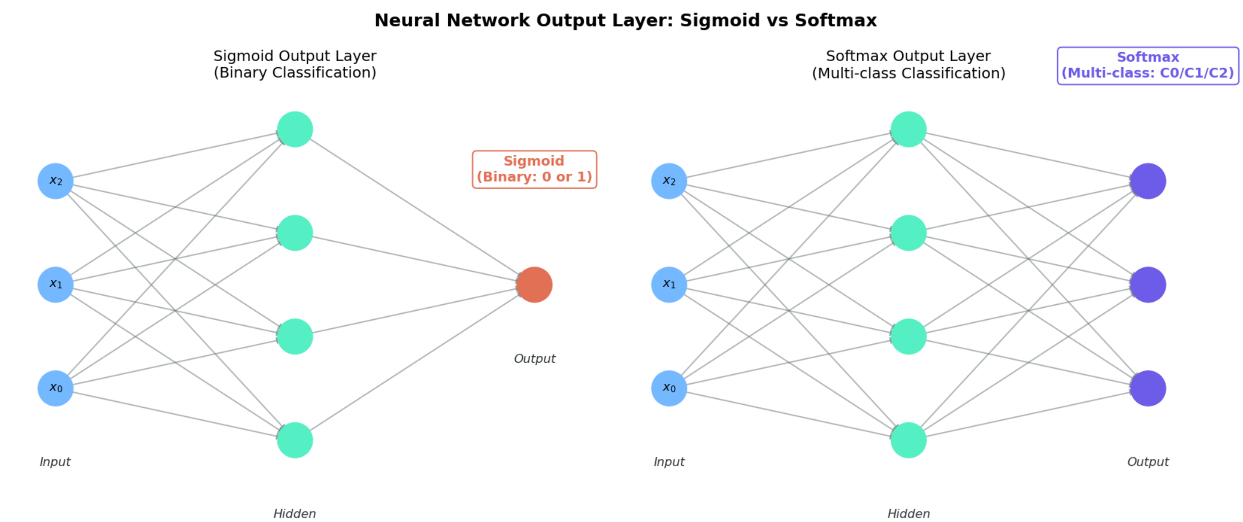
5 Softmax in Neural Networks

Softmax converts raw logits $z = [z_1, \dots, z_k]$ from a neural network into a valid probability distribution over k classes:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}, \qquad \sigma(z)_i \in [0,1], \quad \sum_i \sigma(z)_i = 1$$

Sigmoid vs. Softmax: Output Layer Design

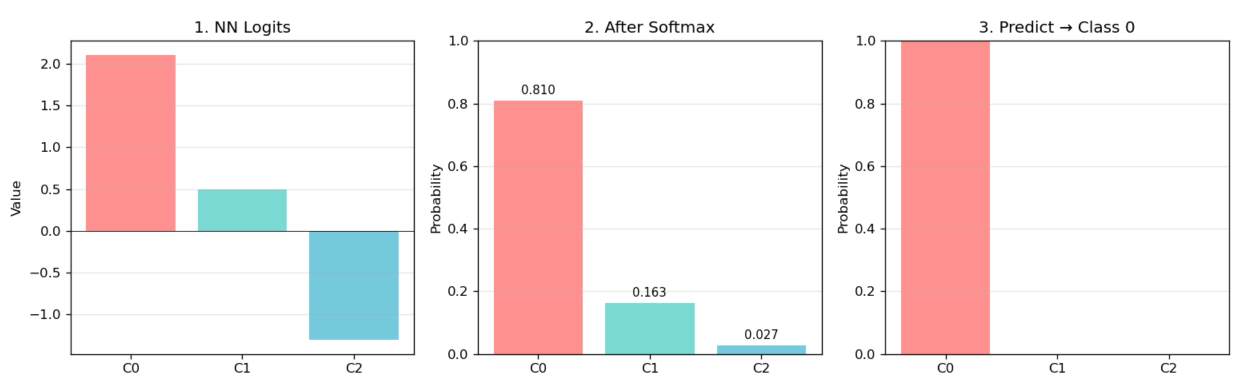
The key architectural difference between binary and multi-class networks is the output layer activation:



- **Sigmoid (left).** One output neuron; $\sigma(z) = 1/(1+e^{-z})$. The output is the probability of the positive class. Suitable only for binary problems.
- **Softmax (right).** One output neuron per class; outputs sum to 1. Enables the network to assign a calibrated probability to each of k classes simultaneously.

Softmax is a generalisation of sigmoid: for $k = 2$ the two are mathematically equivalent.

Classification Pipeline



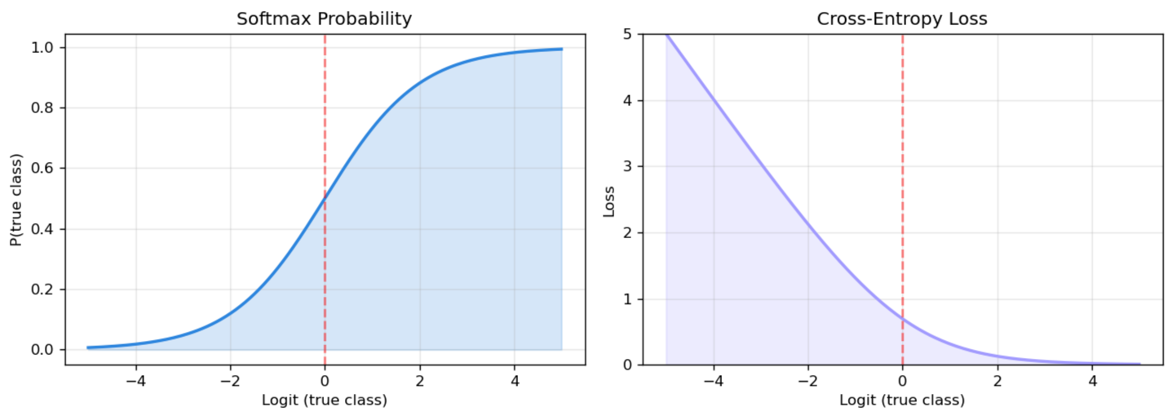
Raw logits (any real number) are exponentiated and normalised by softmax. The predicted class is $\hat{y} = \arg\max_i \sigma(z)_i$.

Training with Cross-Entropy Loss

Softmax is paired with cross-entropy loss during training. Since only the true class y_{true} contributes:

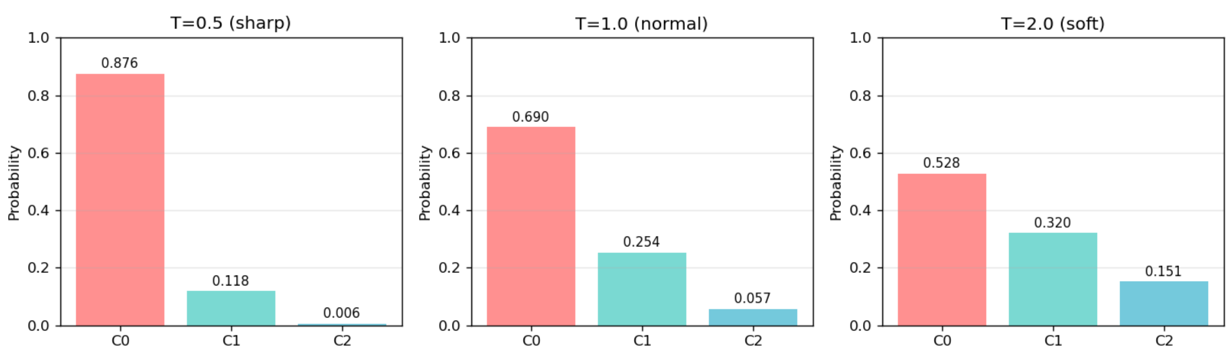
$$\ell = -\log \sigma(z)_{y_{\text{true}}}$$

Loss is high when the model is wrong and shrinks as confidence in the correct class grows.



Temperature Scaling

Dividing logits by temperature T adjusts output sharpness. $T < 1$ sharpens predictions (more confident); $T > 1$ softens them (useful in knowledge distillation where a teacher model's "soft" probabilities guide training of a smaller student model).



6 Conclusion

- **MLE** maximises data likelihood and converges to the true parameters as sample size grows.

- **Method of Moments** offers fast, closed-form estimates by matching moments; preferred when the likelihood is intractable or data is limited.
- **Covariance** reveals feature relationships, guiding feature selection, dimensionality reduction, and model interpretability.
- **KL Divergence** measures distribution differences and underlies cross-entropy loss and generative model training.
- **Softmax** is the standard output activation for multi-class neural networks, converting logits into calibrated class probabilities and enabling training via cross-entropy loss.