Uncertainty in Deep Learning

Yarin Gal

University of Oxford
yarin@cs.ox.ac.uk

Uncertainty over Functions

Structure

- Our model
- Decomposing uncertainty
- Aleatoric uncertainty
- Epistemic uncertainty

(a) Input Image (b) Semantic Segmentation (c) Epistemic Uncertainty

Model

- prior

$$p(w_{k,d}) = \mathcal{N}(w_{k,d}; 0, s^2); \quad W \in \mathbb{R}^{K \times 1}$$

- likelihood

$$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$

- Posterior

$$p(W|X, Y) = \mathcal{N}(W; \mu', \Sigma')$$

$$\Sigma' = (\sigma^{-2} \Phi(X)^T \Phi(X) + s^{-2} I_K)^{-1}$$

$$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$

- Predictive

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T \varphi(x^*), \sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*))$$

Model

- prior

$$p(w_{k,d}) = \mathcal{N}(w_{k,d}; 0, s^2); \quad W \in \mathbb{R}^{K\times 1}$$

- likelihood

$$p(Y|X,W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T\varphi(x)$$

- Posterior

$$p(W|X,Y) = \mathcal{N}(W; \mu', \Sigma')$$
$$\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I_K)^{-1}$$
$$\mu' = \Sigma'\sigma^{-2}\Phi(X)^T Y$$

- Predictive

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T\varphi(x^*), \sigma^2 + \varphi(x^*)^T\Sigma'\varphi(x^*))$$

Model

- prior

$$p(w_{k,d}) = \mathcal{N}(w_{k,d}; 0, s^2); \quad W \in \mathbb{R}^{K \times 1}$$

- likelihood

$$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$

- Posterior

$$p(W|X, Y) = \mathcal{N}(W; \mu', \Sigma')$$
$$\Sigma' = (\sigma^{-2}\Phi(X)^T \Phi(X) + s^{-2} I_K)^{-1}$$
$$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$

- Predictive

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T \varphi(x^*), \sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*))$$

UNIVERSITY OF OXFORD

Model

- prior

$$p(w_{k,d}) = \mathcal{N}(w_{k,d}; 0, s^2); \quad W \in \mathbb{R}^{K \times 1}$$

- likelihood

$$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$

- Posterior

$$\begin{aligned} p(W|X, Y) &= \mathcal{N}(W; \mu', \Sigma') \\ \Sigma' &= (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I_K)^{-1} \\ \mu' &= \Sigma'\sigma^{-2}\Phi(X)^T Y \end{aligned}$$

- Predictive

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T\varphi(x^*), \sigma^2 + \varphi(x^*)^T\Sigma'\varphi(x^*))$$

Model

- prior

$$p(w_{k,d}) = \mathcal{N}(w_{k,d}; 0, s^2); \quad W \in \mathbb{R}^{K \times 1}$$

- likelihood

$$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$

- Posterior

$$p(W|X, Y) = \mathcal{N}(W; \mu', \Sigma')$$
$$\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2} I_K)^{-1}$$
$$\mu' = \Sigma'\sigma^{-2}\Phi(X)^T Y$$

- **Predictive**

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T\varphi(x^*), \overbrace{\sigma^2 + \varphi(x^*)^T\Sigma'\varphi(x^*)}^{\text{predictive uncertainty}})$$

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T \varphi(x^*),$$
$$\sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*))$$

- Variance of predictive dist is the predictive uncertainty
- Uncertainty has two components:
  - σ^2 – from likelihood
    $$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$
  - $\varphi(x^*)^T \Sigma' \varphi(x^*)$ – from posterior
    $$p(W|X, Y) = \mathcal{N}(W; \mu', \Sigma')$$
    $$\Sigma' = (\sigma^{-2} \Phi(X)^T \Phi(X) + s^{-2} I_K)^{-1}$$
    $$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$
- These two terms have very different interpretations

UNIVERSITY OF OXFORD

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T \varphi(x^*),$$
$$\sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*))$$

- Variance of predictive dist is the predictive uncertainty
- Uncertainty has two components:
  - σ^2 – from likelihood
    $$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$
  - $\varphi(x^*)^T \Sigma' \varphi(x^*)$ – from posterior
    $$p(W|X, Y) = \mathcal{N}(W; \mu', \Sigma')$$
    $$\Sigma' = (\sigma^{-2} \Phi(X)^T \Phi(X) + s^{-2} I_K)^{-1}$$
    $$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$
- These two terms have very different interpretations

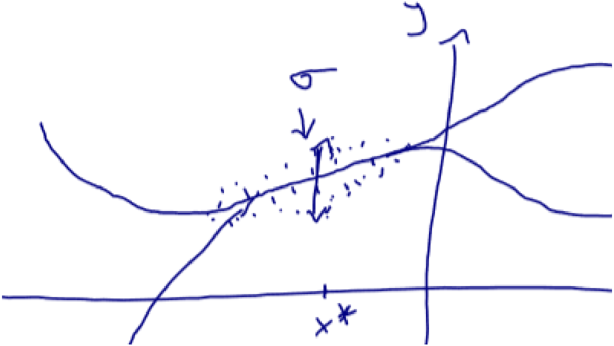
Decoding uncertainty

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \mu'^T \varphi(x^*),$$
$$\sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*))$$

- Variance of predictive dist is the predictive uncertainty
- Uncertainty has two components:
  - σ^2 – from likelihood
    $$p(Y|X, W) = \prod_n \mathcal{N}(y_n; f^W(x_n), \sigma^2); \quad f^W(x) = W^T \varphi(x)$$
  - $\varphi(x^*)^T \Sigma' \varphi(x^*)$ – from posterior
    $$p(W|X, Y) = \mathcal{N}(W; \mu', \Sigma')$$
    $$\Sigma' = (\sigma^{-2} \Phi(X)^T \Phi(X) + s^{-2} I_K)^{-1}$$
    $$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$
- These two terms have very different interpretations

Aleatoric uncertainty

- first term in predictive uncertainty $\sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*)$
- same as likelihood σ^2 – obs noise / corrupting additive noise eg measurement error
- no matter how many training y's we see at x^*, σ^2 will stay the same (it'll actually be the variance of the training y's we see at x)

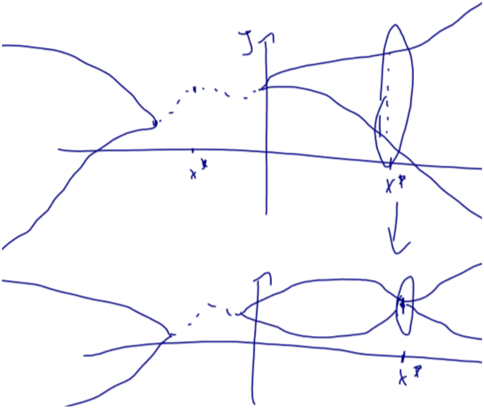


Aleatoric uncertainty

- σ^2 can be found via MLE rather than assumed to be known in advance (we'll see later)
- called 'aleatoric' uncertainty, from Latin aleator 'dice player', from alea 'die'
  - roll a pair of dice again and again – will not reduce uncertainty

Epistemic uncertainty

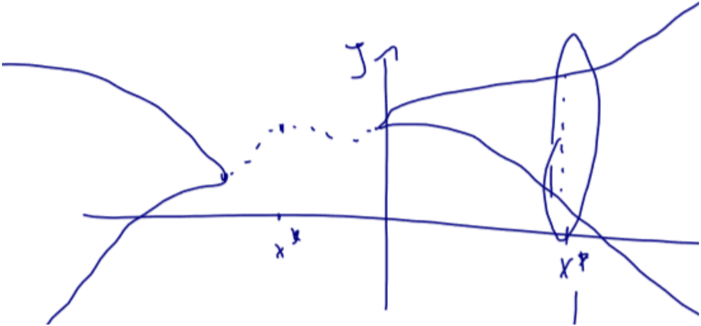
- second term in predictive uncertainty $\sigma^2 + \varphi(x^*)^T \Sigma' \varphi(x^*)$
- as we'll prove below, this will be high for x^* "far away" from the data, even in noiseless case (ie likelihood noise is zero)
- will diminish if we add label for x^* into training set



UNIVERSITY OF OXFORD

- called 'epistemic' uncertainty, from Ancient Greek episteme 'knowledge, understanding'
- mathematically, this is also the same as uncertainty over **function values** before noise corruption;

$$\text{Define } f^* = W^T \varphi(x^*),$$
$$\mathrm{Var}_{p(f^*|x^*,X,Y)}[f^*] = \varphi(x^*)^T \Sigma' \varphi(x^*)$$



Isolating sources of uncertainty (Exercise)

- Definition: Dirac delta $\delta(X = a)$ is a distribution defined as $\int g(X)\delta(X = a)dX = g(a)$ for all functions g
- Alternative generative story to the above: [whiteboard]

$$f_n | x_n, W \sim \delta(f_n = W^T \varphi(x_n))$$
$$y_n | f_n \sim \mathcal{N}(y_n; f_n, \sigma^2)$$

- Exercise: Show that for the new generative story we have

$$\mathrm{Var}_{p(y^*|f^*,X,Y)}[y^*] = \sigma^2$$

and

$$\mathrm{Var}_{p(f^*|x^*,X,Y)}[f^*] = \varphi(x^*)^T \Sigma' \varphi(x^*)$$

(hint: use the identity and $Var(z) = E[z^T z] - E[z]^T E[z]$ with simple manipulations)

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with $\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I_K)^{-1}$

- Large uncertainty when 'far away' from training set:

$$\mathcal{U}(x^*) >> 0$$

  with x^* dissimilar to training x's

- and low uncertainty when 'near' training set:

$$\mathcal{U}(x^*) \approx 0$$

  with x^* similar to training x's

- We'll show this next under some simplifying assumptions.

Epistemic uncertainty intuition

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with $\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I_K)^{-1}$

- Large uncertainty when 'far away' from training set:

  $$\mathcal{U}(x^*) >> 0$$

  with x^* dissimilar to training x's

- and low uncertainty when 'near' training set:

  $$\mathcal{U}(x^*) \approx 0$$

  with x^* similar to training x's

- We'll show this next under some simplifying assumptions.

Epistemic uncertainty intuition

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with $\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I_K)^{-1}$

- Large uncertainty when 'far away' from training set:

  $$\mathcal{U}(x^*) >> 0$$

  with x^* dissimilar to training x's

- and low uncertainty when 'near' training set:

  $$\mathcal{U}(x^*) \approx 0$$

  with x^* similar to training x's

- We'll show this next under some simplifying assumptions.

Epistemic uncertainty intuition

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with $\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I_K)^{-1}$

- Large uncertainty when 'far away' from training set:

  $$\mathcal{U}(x^*) >> 0$$

  with x^* dissimilar to training x's

- and low uncertainty when 'near' training set:

  $$\mathcal{U}(x^*) \approx 0$$

  with x^* similar to training x's

- We'll show this next under some simplifying assumptions.

How do we define 'similar' and 'dissimilar' / 'near' and 'far away'?

- use **inner product** of feature vectors:

$$k(x^*, x) := \varphi(x^*)^T \varphi(x)$$

(assume inner product is postitive semidefinite, ie $k(x^*, x) \geq 0$)

- x's which are 'similar' / 'near by' have a high k value
  - eg when there exists a training point x_n which equals x^* exactly, k will be largest

$$k(x^*, x_n) = k(x_n, x_n) = \varphi(x_n)^T \varphi(x_n) = ||\varphi(x_n)||_2^2$$

- x's which are 'dissimilar' / 'far away' have a low k value
  - eg if two points' feature vectors are orthogonal to each other, they'll have 0 k value

$$k(x^*, x_n) = 0$$

- For simplicity of derivation, assume that all training points are 'far enough' from each other so

$$k(x_m, x_n) \approx 0$$

for $m \neq n$

How do we define 'similar' and 'dissimilar' / 'near' and 'far away'?

- use **inner product** of feature vectors:

$$k(x^*, x) := \varphi(x^*)^T \varphi(x)$$

(assume inner product is postitive semidefinite, ie $k(x^*, x) \geq 0$)

- x's which are 'similar' / 'near by' have a high k value
  - eg when there exists a training point x_n which equals x^* exactly, k will be largest

$$k(x^*, x_n) = k(x_n, x_n) = \varphi(x_n)^T \varphi(x_n) = ||\varphi(x_n)||_2^2$$

- x's which are 'dissimilar' / 'far away' have a low k value
  - eg if two points' feature vectors are orthogonal to each other, they'll have 0 k value

$$k(x^*, x_n) = 0$$

- For simplicity of derivation, assume that all training points are 'far enough' from each other so

$$k(x_m, x_n) \approx 0$$

for $m \neq n$

Epistemic uncertainty intuition

How do we define 'similar' and 'dissimilar' / 'near' and 'far away'?

- use **inner product** of feature vectors:

$$k(x^*, x) := \varphi(x^*)^T \varphi(x)$$

  (assume inner product is postitive semidefinite, ie $k(x^*, x) \geq 0$)
- x's which are 'similar' / 'near by' have a high k value
  - eg when there exists a training point x_n which equals x^* exactly, k will be largest

$$k(x^*, x_n) = k(x_n, x_n) = \varphi(x_n)^T \varphi(x_n) = ||\varphi(x_n)||_2^2$$

- x's which are 'dissimilar' / 'far away' have a low k value
  - eg if two points' feature vectors are orthogonal to each other, they'll have 0 k value

$$k(x^*, x_n) = 0$$

- For simplicity of derivation, assume that all training points are 'far enough' from each other so

$$k(x_m, x_n) \approx 0$$

  for $m \neq n$

Epistemic uncertainty intuition

How do we define 'similar' and 'dissimilar' / 'near' and 'far away'?

- use **inner product** of feature vectors:

$$k(x^*, x) := \varphi(x^*)^T \varphi(x)$$

  (assume inner product is postitive semidefinite, ie $k(x^*, x) \geq 0$)
- x's which are 'similar' / 'near by' have a high k value
  - eg when there exists a training point x_n which equals x^* exactly, k will be largest

$$k(x^*, x_n) = k(x_n, x_n) = \varphi(x_n)^T \varphi(x_n) = ||\varphi(x_n)||_2^2$$

- x's which are 'dissimilar' / 'far away' have a low k value
  - eg if two points' feature vectors are orthogonal to each other, they'll have 0 k value

$$k(x^*, x_n) = 0$$

- For simplicity of derivation, assume that all training points are 'far enough' from each other so

$$k(x_m, x_n) \approx 0$$

  for $m \neq n$

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with

$$\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I)^{-1}$$

- We'll rearrange Σ' a bit and colour-code it: [whiteboard]

$$\Sigma' = \left(s^{-2}(I + \Phi(X)^T(s^2\sigma^{-2}I)\Phi(X))\right)^{-1}$$

- We'll also use the Woodbury matrix identity:

$$(I + UCV)^{-1} = I - U(C^{-1} + VU)^{-1}V$$

with $U = \Phi(X)^T$, $C = s^2\sigma^{-2}I$, $V = \Phi(X)$,

- Together, we get

$$\Sigma' = s^2\left(I - \Phi(X)^T\left((s^2\sigma^{-2}I)^{-1} + \Phi(X)\Phi(X)^T\right)^{-1}\Phi(X)\right)$$

Epistemic uncertainty intuition

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with

$$\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I)^{-1}$$

- We'll rearrange Σ' a bit and colour-code it: [whiteboard]

$$\Sigma' = \left(s^{-2}(I + \Phi(X)^T(s^2\sigma^{-2}I)\Phi(X)) \right)^{-1}$$

- We'll also use the Woodbury matrix identity:

$$(I + UCV)^{-1} = I - U(C^{-1} + VU)^{-1}V$$

with $U = \Phi(X)^T$, $C = s^2\sigma^{-2}I$, $V = \Phi(X)$, **Exercise**:

  ? What are the dims of U, V, and matrix products?

  ? What's the time complexity for the inverses?

- Together, we get

Epistemic uncertainty intuition

Epistemic uncertainty:

$$\mathcal{U}(x^*) := \varphi(x^*)^T \Sigma' \varphi(x^*)$$

with

$$\Sigma' = (\sigma^{-2}\Phi(X)^T\Phi(X) + s^{-2}I)^{-1}$$

- We'll rearrange Σ' a bit and colour-code it: [whiteboard]

$$\Sigma' = \left(s^{-2}(I + \Phi(X)^T(s^2\sigma^{-2}I)\Phi(X)) \right)^{-1}$$

- We'll also use the Woodbury matrix identity:

$$(I + UCV)^{-1} = I - U(C^{-1} + VU)^{-1}V$$

  with $U = \Phi(X)^T$, $C = s^2\sigma^{-2}I$, $V = \Phi(X)$,
- Together, we get

$$\Sigma' = s^2\left(I - \Phi(X)^T \left((s^2\sigma^{-2}I)^{-1} + \Phi(X)\Phi(X)^T \right)^{-1} \Phi(X) \right)$$

Epistemic uncertainty intuition

Next we'll use our new reformulation of Σ' to show that

- if x^* is dissimilar/far away from all training points, ie $k(x^*, x_n) \approx 0$ for all n, then

$$\mathcal{U}(x^*_{\text{far}}) \approx s^2 k(x^*, x^*)$$

- whereas if x^* is similar/near the data (for simplicity, it actually matches one of the data points $x^* = x_m$), then

$$\mathcal{U}(x^*_{\text{near}}) \approx$$
$$s^2 k(x^*, x^*) - s^2 k(x^*, x_m)\left(\sigma^2 s^{-2} + k(x_m, x_m)\right)^{-1} k(x^*, x_m)$$
$$< \mathcal{U}(x^*_{\text{far}})$$

[whiteboard]

Epistemic uncertainty intuition

Next we'll use our new reformulation of Σ' to show that

- if x^* is dissimilar/far away from all training points, ie $k(x^*, x_n) \approx 0$ for all n, then

$$\mathcal{U}(x^*_{\text{far}}) \approx s^2 k(x^*, x^*)$$

- whereas if x^* is similar/near the data (for simplicity, it actually matches one of the data points $x^* = x_m$), then

$$\begin{aligned}
&\mathcal{U}(x^*_{\text{near}}) \approx \\
&s^2 k(x^*, x^*) - s^2 k(x^*, x_m)\big(\sigma^2 s^{-2} + k(x_m, x_m)\big)^{-1} k(x^*, x_m) \\
&< \mathcal{U}(x^*_{\text{far}})
\end{aligned}$$

[whiteboard]

- In our derivation, we reformulated the model's predictive **variance** in terms of the similarity measure $k(x_1, x_2)$

$$\mathcal{N}(y^*; \mu'^T \varphi(x^*),$$
$$s^2 k(x^*, x^*) - s^2 k(x^*, X)(\sigma^2 s^{-2} I + K)^{-1} k(X, x^*))$$
$$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$

with $k(x^*, X) = [k(x^*, x_n)]_n$ and $K = [k(x_m, x_n)]_{mn}$

- as we'll show next, we can go a step further and write the predictive mean in terms of $k(\cdot, \cdot)$ as well:

$$\mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*) - s^2 k(x^*, X)(\sigma^2 s^{-2} I + K)^{-1} k(X, x^*))$$

[whiteboard]

- In our derivation, we reformulated the model's predictive **variance** in terms of the similarity measure $k(x_1, x_2)$

$$\mathcal{N}(y^*; \mu'^T \varphi(x^*),$$
$$s^2 k(x^*, x^*) - s^2 k(x^*, X)(\sigma^2 s^{-2} I + K)^{-1} k(X, x^*))$$
$$\mu' = \Sigma' \sigma^{-2} \Phi(X)^T Y$$

with $k(x^*, X) = [k(x^*, x_n)]_n$ and $K = [k(x_m, x_n)]_{mn}$

- as we'll show next, we can go a step further and write the predictive mean in terms of $k(\cdot, \cdot)$ as well:

$$\mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*) - s^2 k(x^*, X)(\sigma^2 s^{-2} I + K)^{-1} k(X, x^*))$$

[whiteboard]

Digging deeper

Our predictive dist

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X, X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X, X))^{-1} k(X, x^*))$$

- Why did we go through all this effort?
- We can change our model's φ, and the only thing that changes is the definition of the function $k(\cdot, \cdot)$, the predictive stays the same!
- For example, we can increase the number of elements in $\varphi(x)$ (number of units in our neural network) to infinity, and as long as we can still compute $k(\cdot, \cdot)$, we can perform predictions!
- Turns out, for many φ's we can compute $k(\cdot, \cdot)$ even with **infinite** size φ, eg $\varphi(x) = [\cos(w_n \alpha x + b_n)]_{n=1}^{\infty}$ with randomised $w_n, b_n \sim \mathcal{N}$ gives $k(x_1, x_2) = e^{-\frac{1}{2}||\alpha x_1 - \alpha x_2||_2^2}$, the ***RBF kernel***.

Digging deeper

Our predictive dist

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X, X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X, X))^{-1} k(X, x^*))$$

- Why did we go through all this effort?
- We can change our model's φ, and the only thing that changes is the definition of the function $k(\cdot, \cdot)$, the predictive stays the same!
- For example, we can increase the number of elements in $\varphi(x)$ (number of units in our neural network) to infinity, and as long as we can still compute $k(\cdot, \cdot)$, we can perform predictions!
- Turns out, for many φ's we can compute $k(\cdot, \cdot)$ even with **infinite** size φ, eg $\varphi(x) = [\cos(w_n \alpha x + b_n)]_{n=1}^{\infty}$ with randomised $w_n, b_n \sim \mathcal{N}$ gives $k(x_1, x_2) = e^{-\frac{1}{2}||\alpha x_1 - \alpha x_2||_2^2}$, the ***RBF kernel***.

Our predictive dist

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X,X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X,X))^{-1} k(X, x^*))$$

- Why did we go through all this effort?
- We can change our model's φ, and the only thing that changes is the definition of the function $k(\cdot,\cdot)$, the predictive stays the same!
- For example, we can increase the number of elements in $\varphi(x)$ (number of units in our neural network) to infinity, and as long as we can still compute $k(\cdot,\cdot)$, we can perform predictions!
- Turns out, for many φ's we can compute $k(\cdot,\cdot)$ even with **infinite** size φ, eg $\varphi(x) = [\cos(w_n \alpha x + b_n)]_{n=1}^{\infty}$ with randomised $w_n, b_n \sim \mathcal{N}$ gives $k(x_1, x_2) = e^{-\frac{1}{2}||\alpha x_1 - \alpha x_2||_2^2}$, the ***RBF kernel***.

Our predictive dist

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X, X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X, X))^{-1} k(X, x^*))$$

- Why did we go through all this effort?
- We can change our model's φ, and the only thing that changes is the definition of the function $k(\cdot, \cdot)$, the predictive stays the same!
- For example, we can increase the number of elements in $\varphi(x)$ (number of units in our neural network) to infinity, and as long as we can still compute $k(\cdot, \cdot)$, we can perform predictions!
- Turns out, for many φ's we can compute $k(\cdot, \cdot)$ even with **infinite** size φ, eg $\varphi(x) = [\cos(w_n \alpha x + b_n)]_{n=1}^{\infty}$ with randomised $w_n, b_n \sim \mathcal{N}$ gives $k(x_1, x_2) = e^{-\frac{1}{2}||\alpha x_1 - \alpha x_2||_2^2}$, the ***RBF kernel***.

Digging deeper

- We re-formulated our predictive dist in **data space** instead of **feature space**
- this allowed us to gain insight about the decreasing uncertainty near training data
- this model is known as a **Gaussian process** (GP, and $k(\cdot,\cdot)$ is known as a kernel / covariance function). If you want to read more about GPs, see book *Gaussian Processes for Machine Learning*
- GPs can be used to give more insights into neural nets (eg see *"Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs"*)

Digging deeper

- We re-formulated our predictive dist in **data space** instead of **feature space**
- this allowed us to gain insight about the decreasing uncertainty near training data
- this model is known as a **Gaussian process** (GP, and $k(\cdot, \cdot)$ is known as a kernel / covariance function). If you want to read more about GPs, see book *Gaussian Processes for Machine Learning*
- GPs can be used to give more insights into neural nets (eg see *"Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs"*)

Digging deeper

- We re-formulated our predictive dist in **data space** instead of **feature space**
- this allowed us to gain insight about the decreasing uncertainty near training data
- this model is known as a **Gaussian process** (GP, and $k(\cdot, \cdot)$ is known as a kernel / covariance function). If you want to read more about GPs, see book *Gaussian Processes for Machine Learning*
- GPs can be used to give more insights into neural nets (eg see *"Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs"*)

- We re-formulated our predictive dist in **data space** instead of **feature space**
- this allowed us to gain insight about the decreasing uncertainty near training data
- this model is known as a **Gaussian process** (GP, and $k(\cdot,\cdot)$ is known as a kernel / covariance function). If you want to read more about GPs, see book *Gaussian Processes for Machine Learning*
- GPs can be used to give more insights into neural nets (eg see *"Improving the Gaussian Process Sparse Spectrum Approximation by Representing Uncertainty in Frequency Inputs"*)

Digging deeper

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X,X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X,X))^{-1} k(X, x^*))$$

- however, new predictive dist came at the expense of an N by N matrix inversion, which we tried to avoid earlier
- this derivation also relied heavily on identities of Gaussians, and doesn't necessarily work with non-Gaussians likelihoods and priors, or with deeper neural networks
- In the next lecture we'll see an alternative approach to perform inference in our model which will scale better and allow us to work with **deep neural networks** as well.

Digging deeper

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X, X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X, X))^{-1} k(X, x^*))$$

- however, new predictive dist came at the expense of an N by N matrix inversion, which we tried to avoid earlier
- this derivation also relied heavily on identities of Gaussians, and doesn't necessarily work with non-Gaussians likelihoods and priors, or with deeper neural networks
- In the next lecture we'll see an alternative approach to perform inference in our model which will scale better and allow us to work with **deep neural networks** as well.

$$p(y^*|x^*, X, Y) = \mathcal{N}(y^*; \sigma^{-2} Y^T (\sigma^{-2} K(X,X) + s^{-2} I)^{-1} k(X, x^*),$$
$$s^2 k(x^*, x^*)$$
$$- s^2 k(x^*, X)(\sigma^2 s^{-2} I + K(X,X))^{-1} k(X, x^*))$$

- however, new predictive dist came at the expense of an N by N matrix inversion, which we tried to avoid earlier
- this derivation also relied heavily on identities of Gaussians, and doesn't necessarily work with non-Gaussians likelihoods and priors, or with deeper neural networks
- In the next lecture we'll see an alternative approach to perform inference in our model which will scale better and allow us to work with **deep neural networks** as well.

Questions & discussion