UROB 2023 **Jméno:** ____________________
Midterm test
Varianta: A Body ____________________

1. **MLE regrese**: Máte na střeše senzor, který měří fyzikální veličinu $x \in \mathbb{R}^+$ související s rychlostí větru, jako je např. úhlová rychlost větrného mlýnku. Předpokládáme, že pro dané měření x, má rychlost větru $y \in \mathbb{R}^+$ Rayleighovo rozdělení pravděpodobnosti:

$$p(y|x, w) = (y - wx) \exp\big(-(y - wx)^2\big)$$

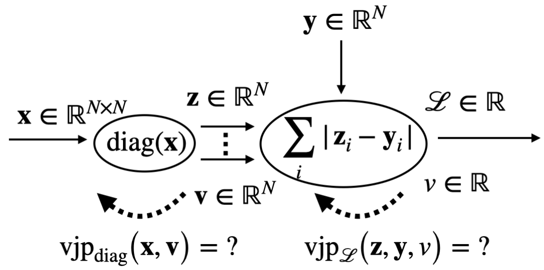
kde $w \in \mathbb{R}$ je neznámý parametr.

- 1.1 Dostanete trénovací sadu $\mathcal{D} = \{(x_1, y_1) \dots (x_N, y_N)\}$ měřených veličin x_i a odpovídajících rychlostí větru y_i. Zapište optimalizační problém, který odpovídá maximálně věrohodnému odhadu parametrů modelu w a pokud je to možné, zjednodušte výsledný optimalizační problém na vhodnou **ztrátovou funkci**.

- 1.2 **Nakreslete výpočetní graf** odvozené ztrátové funkce pro jeden trénovací příklad (x, y) a spočítejte jeho **gradient** vzhledem k w.

- 1.3 Uvažujte nyní, že 1/10 měření nesouvisí se skutečnou rychlostí větru (např. kvůli nějaké vnitřní poruše senzoru). Proto v 1/10 všech tréninkových příkladů pochází rychlost větru y z rovnoměrného rozdělení $y \sim U(0, y_{max})$. **Nakreslete tvar rozdělení pravděpodobnosti**, které modeluje takový případ.

2. **Vector-jacobian-product:** Uvažujte výpočetní graf níže:



Graf se skládá ze dvou funkcí:

a) diag($\mathbf{x}$) funkce, která vrací úhlopříčku vstupní matice $\mathbf{x} \in \mathbb{R}^{N\times N}$ jako sloupcový vektor $\mathbf{y} \in \mathbb{R}^{N\times N}$, který je složený z diagonálních prvků. Například pro $N = 3$, funkce funguje následovně:

$$\mathbf{y} = \text{diag}(\mathbf{x}) = \text{diag}\left(\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}\right) = \begin{bmatrix} x_{11} \\ x_{22} \\ x_{33} \end{bmatrix}$$

b) L1-ztrátová funkce, která je definována jako součet absolutních hodnot rozdílů:

$$\mathcal{L} = \sum_i |\mathbf{z}_i - \mathbf{y}_i|$$

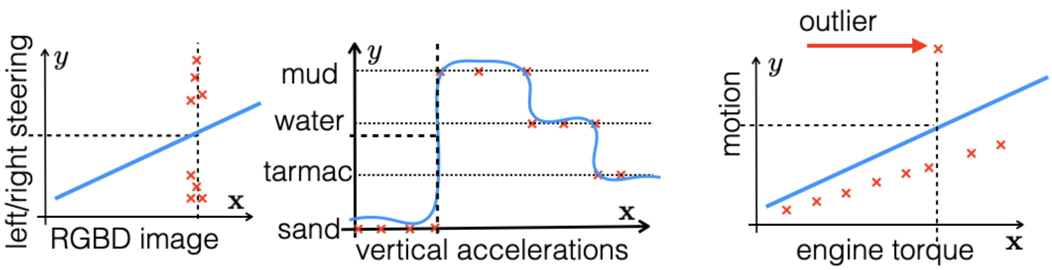
- 2.1 Navrhněte efektivní implementaci funkce vracející **vector-jacobian-product** $\text{vjp}_{\text{diag}}(\mathbf{x}, \mathbf{v})$, kde vektor $\mathbf{v} \in \mathbb{R}^N$ je upstream gradient.

- 2.2 Vypočtěte **gradient** ztrátové funkce $\mathcal{L}$ vzhledem k $\mathbf{x}$ (tj. $N \times N$ matice).

- 2.3 Diskutujte rozdíl mezi vaší efektivní implementací vjp_{diag} a naivním násobením jakobiánů (pouze jedna krátká věta).

3. **Losses and Overfitting:**

- 3.1 Během přednášek jsme diskutovali následující tři problémy, které se objevují při fitování funkce do dat ve smyslu nejmenších čtverců. **Jaký je hlavní zdroj, který tyto problémy způsobuje?** Vyberte právě jednu odpověď.



  - overfitting
  - underfitting
  - výsledný problém je nepříznivý pro optimalizaci (velké plochy s nulovým gradientem)
  - odpovídající tvar ztrátové funkce má ostré minimum, které způsobuje špatné zobecnění testovacích dat
  - model předpokládá, že výstupy y mají normální (Gaussovské) rozdělení pro danou hodnotou $\mathbf{x}$.
  - KL-divergence není definována
  - je velký rozdíl mezi trénovacím/testovacím pravděpodobnostním rozdělením (training/testing distribution mismatch).

- 3.2 **Jak mohu omezit přefitovaní (overfitting)?** Vyberte žádnou, jednu nebo více odpovědí:
  - Nikdy nepředpokládat Gaussovský šum.
  - Použití hodně velké (ideálně nekonečné) datové sady
  - Vyhnout se plochému minimu (flat minimum) ve ztrátové funci
  - Musím udržovat váhy v klasifikátoru/regresoru vždy kladné.
  - Musím udržovat váhy v klasifikátoru/regresoru vždy záporné.
  - Musím použít co nejhlubší konvoluční sítě.
  - Musím použít správný model, který zahrnuje co nejvíc apriorních znalostí o řešeném problému.
  - Vyhnout se ostrému minimu (sharp minimum) ve ztrátové funci.
  - Použít plně diferencovatelnou ztrátovou funkci.
  - Implementovat všechny Vector-Jacobian-Product funkce na GPU.
  - Vždy předpokládat Gaussovský šum.

- 3.3 Označte, které z následujících tvrzení jsou PRAVDA/NEPRAVDA:
  - 2D fitování přímky ve smyslu nejmenších čtverců odpovídá minimalizaci KL-divergence mezi skutečným rozdělením dat p_{data} a normálním (Gaussovským) rozdělením s lineárním průměrem zkonstruovaným následovně

    $$p(\mathbf{x}, y|\mathbf{w}) = \mathcal{N}(y;\ w_1 x + w_0,\ \sigma^2)$$

  - Hodnota globálního optima následujícího problému je vždy rovna nule

    $$\min_{\mathbf{w}} D_{KL}\big(p_{\text{data}}(\mathbf{x}, y) \parallel p(\mathbf{x}, y|\mathbf{w})\big) = 0$$

  - Hodnota globálního optima následujícího problému je vždy nezáporná

    $$\min_{\mathbf{w}} D_{KL}\big(p_{\text{data}}(\mathbf{x}, y) \parallel p(\mathbf{x}, y|\mathbf{w})\big) \geq 0$$

  - Konvoluční síť se nikdy nepřefitovává na obrazových datech.
  - $D_{KL}\big(p_{\text{data}}(\mathbf{x}, y) \parallel p(\mathbf{x}, y|\mathbf{w})\big)$ je monotóně rostoucí funkce vah $\mathbf{w}$.
  - $D_{KL}\big(p_{\text{data}}(\mathbf{x}, y) \parallel p(\mathbf{x}, y|\mathbf{w})\big)$ má právě jedno lokální minimum.
  - Gradient konvoluční vrstvy lze spočítat jako konvoluci.
  - Každý vstup do konvoluční vrstvy vždy ovlivňuje všechny pixely ve výstupní příznakové mapě (feature map).

4. **Convolution:** Předpokládejme, že vstupní příznaková mapa (feature map) do konvoluční vrstvy je $3 \times 10 \times 10$ (channels$\times$height$\times$width).

- 4.1 Jaká je velikost a počet konvolučních jader/filtrů (kernels/filters), pokud by výstup měl být $10 \times 3 \times 3$? (předpokládejme stride=1, padding=0 a dilation_rate=1)

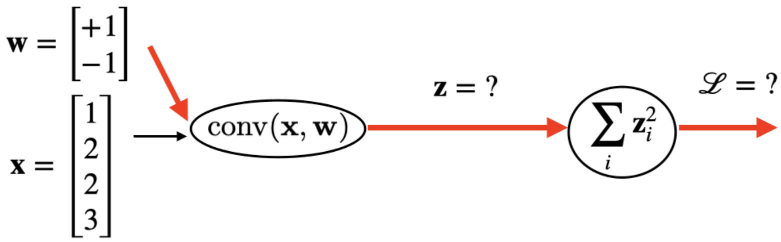
- 4.2 Jaký je padding konvoluční vrstvy, pokud je velikost jádra/filtru $3 \times 3 \times 3$ a výstup by měl být $7 \times 12 \times 12$ (stride=1 a dilation_rate = 1)?

- 4.3 Uvažujme nyní jednokanálový vstup $1 \times 10 \times 10$. Mějme síť sestávající se ze dvou konvolučních vrstev (bez jakékoli aktivační funkce). Každá vrstva má pouze jedno jádro 3×3 následujícího tvaru:

$$\mathbf{w} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}, \ \mathbf{v} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33}, \end{bmatrix}$$

  Můžete tyto dvě vrstvy nahradit plně propojenou vrstvou (fully-connected layer)? Pokud je to možné, jaké budou její váhy? Pokud to není možné, zdůvodněte.

- 4.4 Uvažte následující výpočetní graf s 1D konvolucí. Vypočtěte $\frac{\partial \mathbf{z}}{\partial \mathbf{w}}$ a $\frac{\partial \mathcal{L}}{\partial \mathbf{w}}$:



$\frac{\partial \mathbf{z}}{\partial \mathbf{w}} =$

$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} =$

5. Napište nám cokoliv co se vám zatím libilo/nelíbilo a jak bychom to měli změnit. Pokud máte stále dost času, nakreslete nám obrázek na téma **"Učení robotů a Halloween"**. Vybraná díla budou zveřejněna na stránkách předmětu, nejkreativnější výtvor bude **odměněn lahváčem**.