im trying to show bounds on the trajectory lengths of a neural network with quadratic activatoins using spectral norms of the matrices in the function, but getting stuck when i need to evaluate the integral and relate the input trajectory length to output trajectory length, how to relate them

ChatGPT said:

Thought for 20s

You relate them entirely through the Jacobian and the operator norm inequality. The integral is actually the easy part once that's clear.

Let me write it cleanly and then plug in the quadratic network structure.

---

1. Setup: input vs output trajectory lengths

Let

- x:[0,1]→Rnx:[0,1]\to\mathbb{R}^nx:[0,1]→Rn be a C1C^1C1 path,
- f:Rn→Rmf:\mathbb{R}^n\to\mathbb{R}^mf:Rn→Rm be your network,
- y(t)=f(x(t))y(t) = f(x(t))y(t)=f(x(t)).

Define trajectory length

l(x) := ∫01 // x′(t) // 2 dt,l(f(x)) := ∫01 // y′(t) // 2 dt.l(x) \;:=\; \int_0^1 \big\|x'(t)\big\|_2\,dt, \qquad l(f(x)) \;:=\; \int_0^1 \big\|y'(t)\big\|_2\,dt.l(x):=∫01x′(t)2dt,l(f(x)):=∫01y′(t)2dt.

By the chain rule,

y′(t)=ddtf(x(t))=Jf(x(t)) x′(t),y'(t) = \frac{d}{dt} f(x(t)) = J_f(x(t))\,x'(t),y′(t)=dtdf(x(t))=Jf(x(t))x′(t),

where Jf(x)J_f(x)Jf(x) is the Jacobian of fff at xxx.

So

l(f(x)) = ∫01 // Jf(x(t)) x′(t) // 2 dt.l(f(x)) \;=\; \int_0^1 \big\|J_f(x(t))\,x'(t)\big\|_2\,dt.l(f(x))=∫01Jf(x(t))x′(t)2dt.

---

2. Use the spectral norm inequality inside the integral

By definition of the spectral norm (operator norm induced by Euclidean norm),

∥ Av ∥ 2 ≤ ∥ A ∥ 2 ∥ v ∥ 2for all A,v.\|A v\|_2 \;\le\; \|A\|_2 \,\|v\|_2 \quad \text{for all } A,v. ∥ Av ∥ 2≤ ∥ A ∥ 2 ∥ v ∥ 2for all A,v.

Apply this pointwise with A=Jf(x(t))A = J_f(x(t))A=Jf(x(t)) and v=x′(t)v = x'(t)v=x′(t):

∥ Jf(x(t)) x′(t) ∥ 2 ≤ ∥ Jf(x(t)) ∥ 2 ∥ x′(t) ∥ 2.\big\|J_f(x(t))\,x'(t)\big\|_2 \;\le\; \big\|J_f(x(t))\big\|_2 \,\big\|x'(t)\big\|_2.Jf(x(t))x′(t)2≤Jf(x(t))2x′(t)2.

So

l(f(x))=∫01 ∥ Jf(x(t)) x′(t) ∥ 2 dt ≤ ∫01 ∥ Jf(x(t)) ∥ 2 ∥ x′(t) ∥ 2 dt.l(f(x)) = \int_0^1 \big\|J_f(x(t))\,x'(t)\big\|_2\,dt \;\le\; \int_0^1 \big\|J_f(x(t))\big\|_2 \,\big\|x'(t)\big\|_2\,dt.l(f(x))=∫01Jf(x(t))x′(t)2dt≤∫01Jf(x(t))2x′(t)2dt.

---

3. Pull out a supremum to relate the lengths

Define

g(t):= ∥ Jf(x(t)) ∥ 2.g(t) := \big\|J_f(x(t))\big\|_2.g(t):=Jf(x(t))2.

Then g(t)≤sups∈[0,1]g(s)g(t) \le \sup_{s\in[0,1]} g(s)g(t)≤sups∈[0,1]g(s) for every ttt, so

∫01g(t) ∥ x′(t) ∥ 2 dt ≤ (supt∈[0,1]g(t))∫01∥x′(t)∥2 dt.\int_0^1 g(t)\,\|x'(t)\|_2\,dt \;\le\; \left(\sup_{t\in[0,1]} g(t)\right) \int_0^1 \|x'(t)\|_2\,dt.∫01g(t) ∥ x′(t) ∥ 2dt≤(t∈[0,1]supg(t))∫01 ∥ x′(t) ∥ 2dt.

In other words,

∫01 ∥ Jf(x(t)) ∥ 2 ∥ x′(t) ∥ 2 dt ≤ (supt∈[0,1]∥Jf(x(t))∥2)l(x).\int_0^1 \big\|J_f(x(t))\big\|_2 \,\big\|x'(t)\big\|_2\,dt \;\le\; \left(\sup_{t\in[0,1]} \big\|J_f(x(t))\big\|_2\right) l(x).∫01Jf(x(t))2x′(t)2dt≤(t∈[0,1]supJf(x(t))2)l(x).

Combining with the previous step gives the **upper bound**

l(f(x)) ≤ (supt∈[0,1]∥Jf(x(t))∥2)l(x).\boxed{ l\big(f(x)\big) \;\le\; \left(\sup_{t\in[0,1]} \big\|J_f(x(t))\big\|_2\right) l(x). }l(f(x))≤(t∈[0,1]supJf(x(t))2)l(x).

That's the key "relating input length to output length via the integral" step: it's literally just applying the operator norm inequality inside the integrand and then bounding the scalar coefficient by its supremum along the path.

---

4. Lower bound via the minimum singular value (if you want it)

If Jf(x(t))J_f(x(t))Jf(x(t)) has full column rank and we care about non-degenerate directions, we can use the smallest singular value σmin(Jf(x(t)))\sigma_{\min}(J_f(x(t)))σmin(Jf(x(t))), which satisfies

∥Av∥2 ≥ σmin(A) ∥v∥2.\|A v\|_2 \;\ge\; \sigma_{\min}(A)\,\|v\|_2.∥Av∥2≥σmin(A)∥v∥2.

Then

∥Jf(x(t)) x′(t)∥2 ≥ σmin(Jf(x(t))) ∥x′(t)∥2,\big\|J_f(x(t))\,x'(t)\big\|_2 \;\ge\; \sigma_{\min}\big(J_f(x(t))\big)\,\big\|x'(t)\big\|_2,Jf(x(t))x′(t)2≥σmin(Jf(x(t)))x′(t)2,

so

l(f(x))=∫01 ∥Jf(x(t)) x′(t)∥2 dt ≥ ∫01σmin(Jf(x(t))) ∥x′(t)∥2 dt.l(f(x)) = \int_0^1 \big\|J_f(x(t))\,x'(t)\big\|_2\,dt \;\ge\; \int_0^1 \sigma_{\min}\big(J_f(x(t))\big)\,\big\|x'(t)\big\|_2\,dt.l(f(x))=∫01Jf(x(t))x′(t)2dt≥∫01σmin(Jf(x(t)))x′(t)2dt.

Now

σmin(Jf(x(t))) ≥ infs∈[0,1]σmin(Jf(x(s))),\sigma_{\min}\big(J_f(x(t))\big) \;\ge\; \inf_{s\in[0,1]} \sigma_{\min}\big(J_f(x(s))\big),σmin(Jf(x(t)))≥s∈[0,1]infσmin(Jf(x(s))),

so

l(f(x)) ≥ (inft∈[0,1]σmin(Jf(x(t))))l(x).\boxed{ l\big(f(x)\big) \;\ge\; \left(\inf_{t\in[0,1]} \sigma_{\min}\big(J_f(x(t))\big)\right) l(x). }l(f(x))≥(t∈[0,1]infσmin(Jf(x(t))))l(x).

So upper and lower bounds are controlled by the **max** and **min** singular values of the Jacobian along the trajectory.

---

5. Plug in the quadratic network structure

For a depth-hhh quadratic network (no biases, just to keep the notation clean),

u(1)=W1x,ϕ(u(1))=(u(1))⊙2,u(2)=W2ϕ(u(1)),ϕ(u(2))=(u(2))⊙2,⋮f(x)=Whϕ(u(h−1)),\begin{aligned} u^{(1)} &= W_1 x, \quad &&\phi(u^{(1)}) = (u^{(1)})^{\odot 2},\\ u^{(2)} &= W_2 \phi(u^{(1)}), \quad &&\phi(u^{(2)}) = (u^{(2)})^{\odot 2},\\ &\vdots \\ f(x) &= W_h \phi(u^{(h-1)}), \end{aligned}u(1)u(2)f(x)=W1x,=W2ϕ(u(1)),⋮=Whϕ(u(h−1)),ϕ(u(1))=(u(1))⊙2,ϕ(u(2))=(u(2))⊙2,

the Jacobian can be written as

Jf(x)=WhDh−1(x)Wh−1Dh−2(x)⋯D1(x)W1,J_f(x) = W_h D_{h-1}(x) W_{h-1} D_{h-2}(x) \cdots D_1(x) W_1,Jf(x)=WhDh−1(x)Wh−1Dh−2(x)⋯D1(x)W1, where each Dk(x)D_k(x)Dk(x) is diagonal with entries

Dk(x)=diag (2u1(k)(x),…,2uk-layer width(k)(x)).D_k(x) = \operatorname{diag}\!\big(2 u^{(k)}_1(x),\dots, 2 u^{(k)}_{k\text{-layer width}}(x) \big).Dk(x)=diag(2u1(k)(x),…,2uk-layer width(k)(x)).

Then, using submultiplicativity of the spectral norm,

∥Jf(x)∥2 ≤ ∥Wh∥2∏k=1h−1 ∥Dk(x)∥2 ∥Wk∥2.\|J_f(x)\|_2 \;\le\; \|W_h\|_2 \prod_{k=1}^{h-1} \|D_k(x)\|_2 \,\|W_k\|_2.∥Jf(x)∥2≤∥Wh∥2k=1∏h−1∥Dk(x)∥2∥Wk∥2.

But for a diagonal matrix,

∥Dk(x)∥2=maxi|2ui(k)(x)|=2maxi|ui(k)(x)|.\|D_k(x)\|_2 = \max_i |2 u^{(k)}_i(x)| = 2\max_i |u^{(k)}_i(x)|.∥Dk(x)∥2=imax∣2ui(k)(x)∣=2imax∣ui(k)(x)∣.

So along the trajectory x(t)x(t)x(t),

∥Jf(x(t))∥2 ≤ σmax(Wh)∏k=1h−1(2 σmax(Wk) maxi|ui(k)(t)|),\|J_f(x(t))\|_2 \;\le\; \sigma_{\max}(W_h) \prod_{k=1}^{h-1} \Big(2\, \sigma_{\max}(W_k)\, \max_i |u^{(k)}_i(t)| \Big),∥Jf(x(t))∥2≤σmax(Wh)k=1∏h−1(2σmax(Wk)imax∣ui(k)(t)∣),

and plugging this into the general bound gives

l(f(x)) ≤ supt∈[0,1](σmax(Wh)∏k=1h−12 σmax(Wk) maxi|ui(k)(t)|) l(x).\boxed{ l\big(f(x)\big) \;\le\; \sup_{t\in[0,1]} \left(\sigma_{\max}(W_h) \prod_{k=1}^{h-1} 2\, \sigma_{\max}(W_k)\, \max_i |u^{(k)}_i(t)| \right) \, l(x). }l(f(x))≤t∈[0,1]sup(σmax(Wh)k=1∏h−12σmax(Wk)imax∣ui(k)(t)∣)l(x).

Similarly, if all ui(k)(t)u_i^{(k)}(t)ui(k)(t) stay away from zero and you care about a lower bound, you can use σmin(Wk)\sigma_{\min}(W_k)σmin(Wk) and mini|ui(k)(t)|\min_i |u_i^{(k)}(t)|mini∣ui(k)(t)∣ to get

l(f(x)) ≥ inft∈[0,1](σmin(Wh)∏k=1h−12 σmin(Wk) mini|ui(k)(t)|) l(x).l\big(f(x)\big) \;\ge\; \inf_{t\in[0,1]} \left(\sigma_{\min}(W_h) \prod_{k=1}^{h-1} 2\, \sigma_{\min}(W_k)\, \min_i |u^{(k)}_i(t)| \right) \, l(x).l(f(x))≥t∈[0,1]inf(σmin(Wh)k=1∏h−12σmin(Wk)imin∣ui(k)(t)∣)l(x).

---

6. Intuition in one sentence

- The **Jacobian** tells you the instantaneous expansion/shrinkage of the trajectory.
- The **spectral norm** upper bound says: "at each point, the speed can grow by at most ∥Jf∥2\|J_f\|_2∥Jf∥2, so overall length grows by at most the worst-case expansion along the path, times the original length."

If you paste your current LaTeX inequality where you're stuck with the integral, I can rewrite it directly into this "sup times $I(x)$" form to match your notation exactly.

You said:

what is the supremum

ChatGPT said:

In this context, the **supremum** is just the *least upper bound* of a set of numbers.

More concretely:

1. Formal idea (on $\mathbb{R}$)

Given a set $S \subset \mathbb{R}$, the **supremum** of S, written $\sup S$, is a number L such that:

1. L is an **upper bound**:
   $\forall s \in S,\quad s \le L.$
2. L is the **smallest** such upper bound:
   if U is any other upper bound (so $\forall s \in S,\ s \le U$), then
   $L \le U.$

If the maximum exists, then

$\sup S = \max S.$

But $\sup S$ can also exist even when the max does **not**.

---

2. Simple examples

- Example 1: $S = (0,1)$ (open interval)
  - There is no element of S equal to 1, so $\max S$ doesn't exist.
  - But all elements are ≤ 1, and any smaller number <1 fails to bound the set (since there are elements arbitrarily close to 1).
  - So $\sup S = 1$.
- Example 2: $S = [0,1]$ (closed interval)

- Here maxS=1\max S = 1maxS=1 *and* supS=1\sup S = 1supS=1.

So: **supremum = max if the max exists; otherwise it's the "limit" of the set from above.**

---

3. In your trajectory length bound

When we write something like

supt∈[0,1]∥Jf(x(t))∥2,\sup_{t \in [0,1]} \big\|J_f(x(t))\big\|_2,t∈[0,1]supJf(x(t))2,

we're taking the set

S={∥Jf(x(t))∥2:t∈[0,1]}S = \left\{\big\|J_f(x(t))\big\|_2 : t \in [0,1]\right\}S={Jf(x(t))2:t∈[0,1]}

and asking:

> What is the smallest number LLL such that
> ∥Jf(x(t))∥2≤L\big\|J_f(x(t))\big\|_2 \le LJf(x(t))2≤L for **all** t∈[0,1]t \in [0,1]t∈[0,1]?

Intuitively:

- It's the **worst-case expansion factor** of your network along that trajectory.
- Even if ∥Jf(x(t))∥2\|J_f(x(t))\|_2∥Jf(x(t))∥2 doesn't actually *hit* this value exactly at any specific ttt, the supremum is the tightest possible upper bound that works for all ttt.

Then the inequality

l(f(x))≤(supt∈[0,1]∥Jf(x(t))∥2)l(x)l(f(x)) \le \left(\sup_{t\in[0,1]} \big\|J_f(x(t))\big\|_2\right) l(x)l(f(x))≤(t∈[0,1]supJf(x(t))2)l(x)

is saying:

> The output length is at most "**worst-case local stretching** along the path" ×\times× "original length".

If you want, I can also mirror this with the **infimum** (greatest lower bound) to explain the lower bound version too.

You said:
why do the opposite inequalities for the lower bound apply when we take singular values of the weight matrices/function matrices in the jacobian

ChatGPT said:

Great question — and the answer is subtle but fundamental.
The **upper bound** uses **operator norm inequalities** (which involve **largest singular values**), while the **lower bound** uses **reverse inequalities** for **smallest singular values**. The reason the inequalities "flip direction" is because the largest singular value gives the *maximal* stretching of a transformation, while the smallest singular value gives the *minimal* stretching — and these two behave oppositely.

Let's explain carefully.

✅ 1. The key facts about singular values

For any matrix AAA:

- **Largest singular value (spectral norm)**:
  // A // 2=σmax(A)\|A\|_2 = \sigma_{\max}(A) // A // 2=σmax(A)
  satisfies
  // Av // 2 ≤ σmax(A) ‖v‖2.\|A v\|_2 \;\le\;
  \sigma_{\max}(A)\,\|v\|_2. // Av // 2≤σmax(A) // v // 2.
- **Smallest singular value**:
  σmin(A)\sigma_{\min}(A)σmin(A)
  satisfies
  // Av // 2 ≥ σmin(A) ‖v‖2.\|A v\|_2 \;\ge\;
  \sigma_{\min}(A)\,\|v\|_2. // Av // 2≥σmin(A) // v // 2.

These are *dual inequalities*, not symmetric.

Why?

Because:

- σmax\sigma_{\max}σmax describes the **maximum stretching factor** of the linear transformation.

- σmin\sigma_{\min}σmin describes the **minimum stretching factor** in any direction.

So they must produce **opposite inequalities**.

✅ 2. How this interacts with the Jacobian

Recall your structure:

$$J_f(x) = W_h D_{h-1}(x) W_{h-1} \cdots D_1(x) W_1.$$

For the **upper bound**, you use:

$$\|AB\|_2 \le \|A\|_2 \|B\|_2.$$

So:

$$\|J_f(x)\|_2 \;\le\; \|W_h\|_2 \prod_{k=1}^{h-1} \|D_k(x)\|_2 \|W_k\|_2$$

and because $\|W_k\|_2 = \sigma_{\max}(W_k)$, this creates the **upper bound**.

✅ 3. Why the inequalities reverse for the lower bound

For the **lower bound**, we use the fact that for *nondegenerate matrices*:

$$\sigma_{\min}(AB) \;\ge\; \sigma_{\min}(A)\,\sigma_{\min}(B).$$

This is because:

$$\|AB v\|_2 \;\ge\; \sigma_{\min}(A)\|Bv\|_2 \;\ge\; \sigma_{\min}(A)\sigma_{\min}(B)\|v\|_2.$$

Thus:

$$\sigma_{\min}(AB) \ge \sigma_{\min}(A)\sigma_{\min}(B).$$

Applying this repeatedly:

$$\sigma_{\min}(J_f(x)) \ge \sigma_{\min}(W_h)\prod_{k=1}^{h-1} \sigma_{\min}(D_k(x)) \sigma_{\min}(W_k).$$

And since a diagonal matrix has:

$$\sigma_{\min}(D_k(x)) = \min_i |2 u_i^{(k)}(x)|.$$

✅ 4. Then inside the trajectory integral we use:

Upper bound:

$$\|J_f(x(t))x'(t)\|_2 \le \|J_f(x(t))\|_2 \,\|x'(t)\|_2.$$

Lower bound:

$$\|J_f(x(t))x'(t)\|_2 \ge \sigma_{\min}(J_f(x(t))) \,\|x'(t)\|_2.$$

So the two integrals behave oppositely:

$$l(f(x)) = \int_0^1 \|J_f(x(t))x'(t)\|\,dt$$

Upper bound:

$$\le \left(\sup_t \|J_f(x(t))\|_2\right) \, l(x)$$

Lower bound:

$$\ge \left(\inf_t \sigma_{\min}(J_f(x(t)))\right) \, l(x)$$

The “directions” of the inequalities come from the fact that one bounds growth and the other bounds shrinkage.

✅ 5. Intuition: Stretching vs squeezing

Think of the Jacobian as a linear transformation that stretches and squeezes space differently in different directions:

Upper bound (largest singular value)

“How much can this transform stretch the curve in the *worst* direction?”

Lower bound (smallest singular value)

“How much must this transform stretch the curve in the *least* stretched direction?”

Thus:

- σmax\sigma_{\max}σmax multiplies into an **upper bound**.
- σmin\sigma_{\min}σmin multiplies into a **lower bound**.

The inequalities must flip because:

- σmax\sigma_{\max}σmax represents the **largest** possible magnification factor.
- σmin\sigma_{\min}σmin represents the **smallest** possible magnification factor.

✅ 6. Key summary line

The opposite inequalities arise because:

> **Largest singular values control the maximum possible expansion**, so they give “≤\le≤” bounds.
> **Smallest singular values control the minimum possible expansion (or contraction)**, so they give “≥\ge≥” bounds.

This is how we get both sides of the trajectory length inequality.

If you want, I can write this in LaTeX exactly in the style of your paper’s bound, with the spectral norms and the min/max of the hidden units placed correctly.

You said:

what paper described this notion of expressivity? the topology of $(h \cdot f)^{-1} (e_i) = \{ x \in \RR^n_0: h \circ f(x) = e_i\}$ after composing the network with $h(x) = e_i$ and taking $x_i = \max_j \{x_j\}$ (Limet et al.,

ChatGPT said:

Good question. There **are** papers that study the *topological* expressivity / capacity of neural networks along exactly those lines; some seem quite related to the idea you describe (preimages, decision-boundaries, topology of level-/preimage-sets, etc.). Below are the main ones, and how well they match your description — with a note on why your "$(h \circ f)^{-1}(e_i)$ after a max / argmax or pooling-like map" idea may or may not match exactly what they do.

✅ Relevant papers on topological expressivity

Topology of deep neural networks (by Naitzat, Zhitnikov & Lek-Heng Lim, 2020)

- This paper studies how a data manifold M=Ma∪Mb⊂RdM = M_a \cup M_b \subset \mathbb{R}^dM=Ma∪Mb⊂Rd — representing two classes in a binary classification problem — changes topology as it is mapped by a trained neural network. JMLR+1

- Concretely: They examine how the Betti numbers (topological invariants, roughly counting connected components, holes, loops, etc.) evolve under the network. Their empirical finding: often a complicated topology simplifies drastically, implying the network "untangles" the data topologically. JMLR

- This is very close in spirit to your notion of looking at preimage sets under a composed map, though they focus on classification and on how data topology evolves, rather than on preimages of "max + indicator" functions per se.

Topological Expressivity of ReLU Neural Networks (by Ekin Ergen & Moritz Grillo, 2024)

- More recent, this work gives *formal lower and upper bounds* (in terms of architectural parameters like depth/width) on how much a ReLU network can simplify (or change) the topology of data in a binary classification setting. They measure topological changes via Betti numbers. Proceedings of Machine Learning Research+1

- Their results show that deep networks can perform exponentially greater topological simplification than shallow ones. [Proceedings of Machine Learning Research]
- This is perhaps the closest formal work to a theoretical “capacity” or “expressivity” notion defined via topological transformations, which may align with your interest in (h◦f)−1(ei)(h \circ f)^{-1}(e_i)(h◦f)−1(ei)-type preimages (though again they do not necessarily use a max/argmax map as “hhh”).

On The Topological Expressive Power of Neural Networks (by Giovanni Petri & António Leitão, around 2020)

- This earlier work proposes a “topological description of neural network expressive power” in the sense of the space of decision boundaries (for classification) realized by a network architecture. That is, they study how many “different topologies” of decision boundary a network architecture can represent, by sampling across parameter space. [OpenReview]
- Their approach is more abstract: rather than focusing on a particular preimage set for a fixed fff, they consider the space of all decision boundaries the architecture can realize — and measure the “manifold spread” (in some topological sense) of that space. [OpenReview]
- So if your question is about “how expressive is this architecture, in terms of possible topologies of preimage / decision sets,” this is a very relevant paper.

⚠️ On whether these match exactly “(h◦f)−1(ei)(h \circ f)^{-1}(e_i)(h◦f)−1(ei) after max/arg-max / pooling”

- None of the papers above seems to use exactly the same formalism you described (i.e., h(x)=eih(x) = e_ih(x)=ei where xi=maxjxjx_i = \max_j x_jxi=maxjxj). Instead they usually consider classification networks where the final output is a scalar or low-dimensional — and they study the preimage of class-labels / decision-regions (e.g., f−1((−∞,0])f^{-1}((-\infty, 0])f−1((−∞,0]) vs f−1((0,∞))f^{-1}((0, \infty))f−1((0,∞))), not necessarily a “max + indicator” map hhh.
- The idea of using a “max” or “argmax” layer is more common in architectures with a “max/pooling / maxout” activation, or in soft-max + argmax — but most formal topology-expressivity analyses stick to simpler output codings. For example, the “Topological Expressivity of ReLU Neural Networks” paper is about binary classification

(so effectively a threshold on one real output). [Proceedings of Machine Learning Research+1]

- That said: the conceptual motivation behind those works — quantifying how “topologically complicated” a network’s preimage/decision-region capabilities are — is extremely close to what you want. You might try to generalize their techniques, modifying them to the non-standard “max+one-hot” output case (i.e., your h∘fh \circ fh∘f).

🧠 My take: the best-fitting prior work & what’s novel in your idea

- The 2024 Ergen & Grillo “Topological Expressivity of ReLU Neural Networks” seems like the **state of the art** for formal topological expressivity bounds.
- The Petri & Leitão “On The Topological Expressive Power of Neural Networks” offers a complementary, architectural-level viewpoint: how expressive a **class** of networks is in terms of possible decision-boundary topologies, rather than for a fixed network instance.
- Your idea — analyzing the topology of (h∘f)−1(ei)(h \circ f)^{-1}(e_i)(h∘f)−1(ei) where hhh picks out a “max-class index” — possibly extends those frameworks in an interesting new direction. To my knowledge, **no** paper uses exactly that composition with a “max + one-hot” hhh and analyzes the topology of the resulting preimages in a general expressivity bound. That could be a novel angle.

🔎 Recommendation for you

Since your setting (quadratic activations, potentially “max-or-argmax” output) is somewhat nonstandard vs the usual ReLU / binary-classification in the literature, I suggest:

1. Start from the method of **Ergen & Grillo (2024)**: read how they bound topological simplification via Betti numbers. Try to adapt that method to your activation / output mapping.
2. Use the framework of **Petri & Leitão (≈ 2020)** to think about the space of decision-boundary topologies your architecture can realize — that might give a lower bound on how “diverse” preimage topologies are.

3. Be ready for new technical obstacles: e.g. diagonal Jacobians, non-piecewise-linear activations (quadratic), the fact that “max + one-hot” is not smooth — you may need a variant of their arguments or even new topological tools (stratified sets, singularities, etc.).

If you like — I can **scan the literature (last 5–10 years)** and try to find **all papers** that use *exactly* the kind of preimage-topology / “max-class index + preimage set” expressivity definition you suggest.
Do you want me to build that list for you now?

You said:

what are canonical ml datasets of different modes like text and images etc

ChatGPT said:

Here’s a clean, **mode-by-mode list of canonical machine-learning datasets**, organized the way research papers usually categorize them. I’ll include the most historically important, the most widely used benchmarks today, and what each is typically used for.

🖼️ IMAGE DATASETS

Classification

- **MNIST** — handwritten digits (28×28 grayscale)
- **Fashion-MNIST** — clothing images (replacement for MNIST)
- **CIFAR-10 / CIFAR-100** — small natural images (32×32 color)
- **ImageNet (ILSVRC)** — 1.2M natural images, 1000 classes (core vision benchmark)
- **TinyImageNet** — 64×64 subset of ImageNet (student projects)

Medical Imaging

- **CheXpert / NIH ChestX-ray14** — chest radiographs

- **ISIC** — dermoscopy images for melanoma detection
- **BraTS** — brain tumor MRI segmentation

Segmentation / Detection

- **COCO** — object detection, segmentation, captioning
- **PASCAL VOC** — earlier detection and segmentation benchmark
- **Cityscapes** — street-scene segmentation (autonomous driving)
- **ADE20K** — scene parsing

📝 TEXT / NLP DATASETS

General NLP tasks

- **Penn Treebank (PTB)** — classic LM benchmark
- **IMDb** — sentiment classification
- **AG News** — news topic classification
- **SST-2 / SST-5** — fine-grained sentiment analysis

Large-scale language modeling / pretraining

- **WikiText-103** — large Wikipedia LM corpus
- **The Pile** — 825GB curated text dataset (modern foundation-model pretraining)
- **Common Crawl** — web-scale corpus behind GPT/BERT-style pretraining

Question answering

- **SQuAD v1/v2** — span-extraction QA
- **Natural Questions (NQ)** — Google search QA
- **TriviaQA**, **HotpotQA** — multi-hop reasoning

Machine Translation

- **WMT (various years)** — multilingual parallel corpora
- **IWSLT** — speech and text translation

🔉 AUDIO / SPEECH DATASETS

Speech Recognition

- **LibriSpeech** — 1000 hours of read English speech
- **TED-LIUM** — TED talk speech transcripts

Speaker Identification

- **VoxCeleb 1/2**

Audio Classification

- **ESC-50** — environmental sounds
- **UrbanSound8K** — real-world environmental audio clips

🎥 VIDEO DATASETS

Action Recognition

- **Kinetics-400 / Kinetics-700** — large human action video dataset
- **UCF-101** — smaller but widely used
- **HMDB-51** — classic benchmark

Video object tracking

- **OTB**, **VOT**, **LaSOT**

Video captioning

- **MSR-VTT**
- **YouCook2**

🧬 TABULAR DATASETS

Classic ML benchmarks (UCI Repository)

- **Iris** — classic classification
- **Wine Quality**
- **Adult Income**
- **Breast Cancer Wisconsin**
- **Bank Marketing**

Large-scale tabular datasets

- **KDD Cup 1998/1999**

- **Higgs Boson** dataset
- **Rossmann Store Sales** (Kaggle)

🌐 MULTIMODAL DATASETS

Image + Text

- **CLIP's datasets (LAION-400M/5B)** — web image–text pairs
- **COCO Captions** — images with captions
- **Flickr30k** — captioned images
- **Visual Question Answering (VQA)** — pairs of images + natural language questions

Video + Text

- **HowTo100M** — YouTube instructional videos + transcripts
- **TVQA** — QA based on TV show clips

🌍 REINFORCEMENT LEARNING DATASETS

Control / Simulation

- **OpenAI Gym classic control** (CartPole, Acrobot, etc.)
- **MuJoCo** locomotion tasks (HalfCheetah, Ant, Hopper, Walker2d)
- **D4RL** — offline RL benchmark (Maze2D, Adroit, AntMaze)

Games

- **Atari 2600 suite** (ALE benchmark)
- **DeepMind Control Suite**

🧪 GRAPHS / NETWORK DATASETS

Node/graph classification

- **Cora, Citeseer, PubMed** — citation network datasets
- **OGB (Open Graph Benchmark)** — realistic graph datasets (molecules, proteins, citation networks, social networks)

Molecules

- **QM9** — small molecule quantum properties
- **ZINC** — graph generation benchmark

📊 TIME SERIES DATASETS

- **UCR Time Series Archive** — large collection of univariate TS
- **ECG5000** — ECG classification
- **Electricity** — forecasting
- **Traffic** — forecasting

📈 GENERATIVE MODELING DATASETS

- **MNIST**, **Fashion-MNIST**, **CIFAR-10** — common for GANs
- **CelebA** — facial images
- **LSUN** — large-scale indoor scenes
- **FFHQ** — high-quality human faces (StyleGAN benchmark)

If you want, I can produce a **short curated list**, a **table**, or a **mode-by-mode “if you’re doing X research, use Y dataset” guide**.